In [1]:
```
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analsysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pa
ndas dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

#!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line
if you haven't completed the Foursquare API lab
import folium # map rendering library

#!conda install -c conda-forge geopy --yes
#!from geopy.geocoders import Nominatim # module to convert an address i
nto latitude and longitude values
print('Libraries imported.')
```

```
Libraries imported.
```

In [2]:
```
#data1 = pd.read_html('https://en.wikipedia.org/wiki/List_of_postal_code
s_of_Canada:_M',skiprows=1)[0]
#print(data1)
```

In [3]:
```
import urllib.request
import bs4 as bs
```

In [4]:
```
source = urllib.request.urlopen('https://en.wikipedia.org/wiki/List_of_p
ostal_codes_of_Canada:_M').read()
soup = bs.BeautifulSoup(source,'lxml')
table = soup.find('table', attrs={'class':'wikitable sortable'})
table_rows = table.find_all('tr')
l = []
for tr in table_rows:
    td = tr.find_all('td')
    row = [tr.text for tr in td]
    l.append(row)
df = pd.DataFrame(l, columns=["Postcode", "Borough", "Neighborhood"])
df.head()
```

Out[4]:

	Postcode	Borough	Neighborhood
0	None	None	None
1	M1A	Not assigned	Not assigned\n
2	M2A	Not assigned	Not assigned\n
3	M3A	North York	Parkwoods\n
4	M4A	North York	Victoria Village\n

In [5]:
```
# Delete the first row with postcode as 0
df = df.iloc[1:]
df.head()
```

Out[5]:

	Postcode	Borough	Neighborhood
1	M1A	Not assigned	Not assigned\n
2	M2A	Not assigned	Not assigned\n
3	M3A	North York	Parkwoods\n
4	M4A	North York	Victoria Village\n
5	M5A	Downtown Toronto	Harbourfront\n

In [6]:
```
df.shape
```

Out[6]: (288, 3)

Only process the cells that have an assigned borough. Ignore cells with a borough that is Not assigned.

In [7]:
```
#Only process the cells that have an assigned borough. Ignore cells with
a borough that is Not assigned.
df.shape
df = df[~df['Borough'].isin(['Not assigned'])]
df.shape
```

Out[7]: (211, 3)

In [8]:
```
df.head()
```

Out[8]:

	Postcode	Borough	Neighborhood
3	M3A	North York	Parkwoods\n
4	M4A	North York	Victoria Village\n
5	M5A	Downtown Toronto	Harbourfront\n
6	M5A	Downtown Toronto	Regent Park\n
7	M6A	North York	Lawrence Heights\n

In [9]:
```
df.sort_values('Postcode', axis=0, ascending=True, inplace=True)
# remove all '\n' values from the dataframe
df = df.replace('\n','', regex=True)
```

In [10]:
```
# rstrip all Neighborhood column
df['Neighborhood'] = df['Neighborhood'].str.strip()
df['Borough'] = df['Borough'].str.strip()
```

In [11]:
```
df.head()
```

Out[11]:

	Postcode	Borough	Neighborhood
12	M1B	Scarborough	Rouge
13	M1B	Scarborough	Malvern
30	M1C	Scarborough	Port Union
29	M1C	Scarborough	Rouge Hill
28	M1C	Scarborough	Highland Creek

In [12]:
```
df = df.replace('\n','', regex=True)
df.head()
```

Out[12]:

	Postcode	Borough	Neighborhood
12	M1B	Scarborough	Rouge
13	M1B	Scarborough	Malvern
30	M1C	Scarborough	Port Union
29	M1C	Scarborough	Rouge Hill
28	M1C	Scarborough	Highland Creek

More than one neighborhood can exist in one postal code area

In [13]:
```
torantodf = pd.DataFrame(columns=['Postcode', 'Borough', 'Neighborhood
'])
prev_postcode = ''
prev_neighborhood = ''
prev_borough = ''
unssigned = 'Not assigned'
for index, row in df.iterrows():
    if(row['Postcode'] == prev_postcode):
        prev_neighborhood = prev_neighborhood + ',' + row['Neighborhood
']
    else:
        # copy to new df
        torantodf.loc[len(torantodf)] = [prev_postcode,prev_borough ,pre
v_neighborhood]
        prev_postcode = row['Postcode']
        prev_borough = row['Borough']
        prev_neighborhood = row['Neighborhood']

torantodf.shape
```

Out[13]:
```
(103, 3)
```

In [14]:
```
# Delete the first row with postcode as blank
torantodf = torantodf.iloc[1:]
torantodf.head(20)
```

Out[14]:

	Postcode	Borough	Neighborhood
1	M1B	Scarborough	Rouge,Malvern
2	M1C	Scarborough	Port Union,Rouge Hill,Highland Creek
3	M1E	Scarborough	Guildwood,Morningside,West Hill
4	M1G	Scarborough	Woburn
5	M1H	Scarborough	Cedarbrae
6	M1J	Scarborough	Scarborough Village
7	M1K	Scarborough	East Birchmount Park,Ionview,Kennedy Park
8	M1L	Scarborough	Golden Mile,Oakridge,Clairlea
9	M1M	Scarborough	Cliffcrest,Scarborough Village West,Cliffside
10	M1N	Scarborough	Cliffside West,Birch Cliff
11	M1P	Scarborough	Wexford Heights,Dorset Park,Scarborough Town C...
12	M1R	Scarborough	Maryvale,Wexford
13	M1S	Scarborough	Agincourt
14	M1T	Scarborough	Sullivan,Clarks Corners,Tam O'Shanter
15	M1V	Scarborough	Milliken,Agincourt North,L'Amoreaux East,Steel...
16	M1W	Scarborough	L'Amoreaux West
17	M1X	Scarborough	Upper Rouge
18	M2H	North York	Hillcrest Village
19	M2J	North York	Fairview,Oriole,Henry Farm
20	M2K	North York	Bayview Village

If a cell has a borough but a Not assigned neighborhood, then the neighborhood will be the same as the borough

In [15]:
```
unssigned = 'Not assigned'
mvalue = ''
location = 0
for index, row in torantodf.iterrows():
    if(row['Neighborhood'] == unssigned):
        row['Neighborhood'] = row['Borough']
```

In [16]:
```
torantodf.head(15)
```

Out[16]:

	Postcode	Borough	Neighborhood
1	M1B	Scarborough	Rouge,Malvern
2	M1C	Scarborough	Port Union,Rouge Hill,Highland Creek
3	M1E	Scarborough	Guildwood,Morningside,West Hill
4	M1G	Scarborough	Woburn
5	M1H	Scarborough	Cedarbrae
6	M1J	Scarborough	Scarborough Village
7	M1K	Scarborough	East Birchmount Park,Ionview,Kennedy Park
8	M1L	Scarborough	Golden Mile,Oakridge,Clairlea
9	M1M	Scarborough	Cliffcrest,Scarborough Village West,Cliffside
10	M1N	Scarborough	Cliffside West,Birch Cliff
11	M1P	Scarborough	Wexford Heights,Dorset Park,Scarborough Town C...
12	M1R	Scarborough	Maryvale,Wexford
13	M1S	Scarborough	Agincourt
14	M1T	Scarborough	Sullivan,Clarks Corners,Tam O'Shanter
15	M1V	Scarborough	Milliken,Agincourt North,L'Amoreaux East,Steel...

In [17]:
```
torantodf.shape
```

Out[17]:
```
(102, 3)
```

In [18]:
```
#Geospatial_Coordinates.csv
# makes the passed rows header
#df = pd.DataFrame(l, columns=["Postcode", "Borough", "Neighborhood"])
newdf = pd.read_csv("Geospatial_Coordinates.csv")
#newdf = pd.read_csv("Geospatial_Coordinates.csv", header = None)
```

In [19]:
```
newdf.shape
```

Out[19]:
```
(103, 3)
```

In [20]:
```
newdf.head()
```

Out[20]:

	Postcode	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

In [21]:
```
#torantodf.head()
originaldf = torantodf.copy()
originaldf.head()
```

Out[21]:

	Postcode	Borough	Neighborhood
1	M1B	Scarborough	Rouge,Malvern
2	M1C	Scarborough	Port Union,Rouge Hill,Highland Creek
3	M1E	Scarborough	Guildwood,Morningside,West Hill
4	M1G	Scarborough	Woburn
5	M1H	Scarborough	Cedarbrae

In [22]:
```
# define the dataframe columns
column_names = ['Postcode', 'Borough', 'Neighborhood', 'Latitude', 'Long
itude']

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

In [23]:
```
torantodf.sort_values('Postcode', axis=0, ascending=True, inplace=True)
newdf.sort_values('Postcode', axis=0, ascending=True, inplace=True)
```

In [24]:
```
#torantodf = originaldf.copy()
```

In [25]:
```
torantodf.head()
```

Out[25]:

	Postcode	Borough	Neighborhood
1	M1B	Scarborough	Rouge,Malvern
2	M1C	Scarborough	Port Union,Rouge Hill,Highland Creek
3	M1E	Scarborough	Guildwood,Morningside,West Hill
4	M1G	Scarborough	Woburn
5	M1H	Scarborough	Cedarbrae

Merge the Torantodf with the "Geospatial_Coordinates.csv" with postcode as the key

In [26]:
```
# define the dataframe columns
column_names = ['Postcode', 'Borough', 'Neighborhood', 'Latitude', 'Long
itude']

torantodf['g'] = torantodf.groupby('Postcode').cumcount()
newdf['g'] = newdf.groupby('Postcode').cumcount()
torantodf.merge(newdf).drop('g',1)
toranto_table = pd.merge(torantodf,newdf,on=["Postcode", 'g'],how='outer
').drop('g',1)
```

In [27]:
```
# instantiate the dataframe
toranto_table.head()
```

Out[27]:

	Postcode	Borough	Neighborhood	Latitude	Longitude
0	M1B	Scarborough	Rouge,Malvern	43.806686	-79.194353
1	M1C	Scarborough	Port Union,Rouge Hill,Highland Creek	43.784535	-79.160497
2	M1E	Scarborough	Guildwood,Morningside,West Hill	43.763573	-79.188711
3	M1G	Scarborough	Woburn	43.770992	-79.216917
4	M1H	Scarborough	Cedarbrae	43.773136	-79.239476

In [28]:
```
toranto_table.shape
```

Out[28]:
```
(103, 5)
```

In [29]:
```
print('The dataframe has {} boroughs and {} neighborhoods.'.format(
        len(toranto_table['Borough'].unique()),
        toranto_table.shape[0]
    )
)
```

```
The dataframe has 12 boroughs and 103 neighborhoods.
```

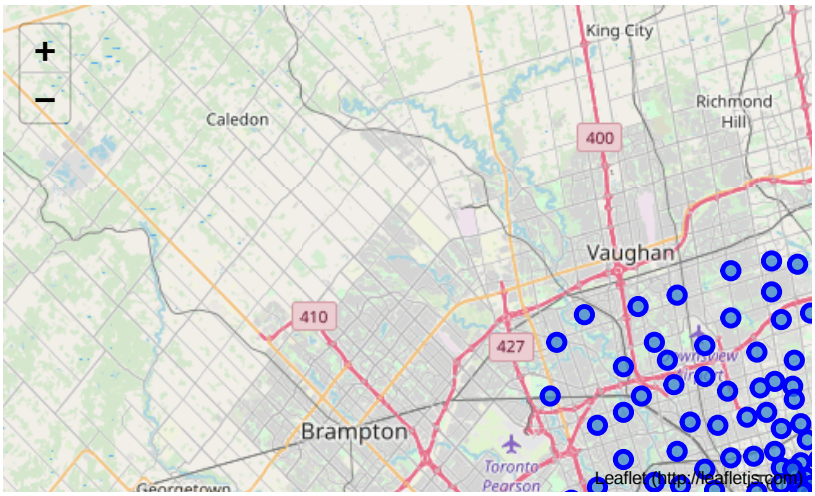
In [30]:
```
#!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim # module to convert an address int
o latitude and longitude values
```

In [31]:
```
address = 'Toronto, Canada'

geolocator = Nominatim(user_agent="toranto_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Toranto, Canada city are {}, {}.'.f
ormat(latitude, longitude))
```

```
The geograpical coordinate of Toranto, Canada city are 43.653963, -79.387
207.
```

In [32]:
```
# create map of Toranto using latitude and longitude values
map_toranto = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, borough, neighborhood in zip(toranto_table['Latitude'], to
ranto_table['Longitude'], toranto_table['Borough'], toranto_table['Neigh
borhood']):
    label = '{}, {}'.format(neighborhood, borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_toranto)

map_toranto
```

Out[32]:



In [33]:
```
CLIENT_ID = 'NMOWZ1J0WTQRB5JYTTXGNCD1BUVCMS0Y0TSNMDZVELYTV4WB' # your Fo
ursquare ID
CLIENT_SECRET = '3ALM5M2BNJGMI1XFR0CC32Z3OMFIHT5V4EZ1JQPH2Y4XE13I' # you
r Foursquare Secret
VERSION = '20180605'
LIMIT = 30
print('Sunil Kelkar :')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Sunil Kelkar :
CLIENT_ID: NMOWZ1J0WTQRB5JYTTXGNCD1BUVCMS0Y0TSNMDZVELYTV4WB
CLIENT_SECRET:3ALM5M2BNJGMI1XFR0CC32Z3OMFIHT5V4EZ1JQPH2Y4XE13I
```

In [34]:
```
toranto_table.loc[0, 'Neighborhood']
```

Out[34]:
```
'Rouge,Malvern'
```

In [35]:
```
neighborhood_latitude = toranto_table.loc[0, 'Latitude'] # neighborhood
latitude value
neighborhood_longitude = toranto_table.loc[0, 'Longitude'] # neighborhoo
d longitude value

neighborhood_name = toranto_table.loc[0, 'Neighborhood'] # neighborhood
name

print('Latitude and longitude values of {} are {}, {}.'.format(neighborh
ood_name,
                                                               neighborh
ood_latitude,
                                                               neighborh
ood_longitude))
```

```
Latitude and longitude values of Rouge,Malvern are 43.806686299999996, -7
9.19435340000001.
```

In [36]:
```
LIMIT = 100 # limit of number of venues returned by Foursquare API

radius = 500 # define radius

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

Out[36]:
```
'https://api.foursquare.com/v2/venues/explore?&client_id=NMOWZ1J0WTQRB5JY
TTXGNCD1BUVCMS0Y0TSNMDZVELYTV4WB&client_secret=3ALM5M2BNJGMI1XFR0CC32Z3OM
FIHT5V4EZ1JQPH2Y4XE13I&v=20180605&ll=43.806686299999996,-79.1943534000000
1&radius=500&limit=100'
```

In [37]:
```
results = requests.get(url).json()
results
```

Out[37]:

```
{'meta': {'code': 200, 'requestId': '5d37e4e36e4650002c441941'},
 'response': {'warning': {'text': "There aren't a lot of results near yo
u. Try something more general, reset your filters, or expand the search a
rea."},
  'headerLocation': 'Malvern',
  'headerFullLocation': 'Malvern, Toronto',
  'headerLocationGranularity': 'neighborhood',
  'totalResults': 2,
  'suggestedBounds': {'ne': {'lat': 43.8111863045, 'lng': -79.18812958073
042},
   'sw': {'lat': 43.80218629549999, 'lng': -79.2005772192696}},
  'groups': [{'type': 'Recommended Places',
    'name': 'recommended',
    'items': [{'reasons': {'count': 0,
       'items': [{'summary': 'This spot is popular',
         'type': 'general',
         'reasonName': 'globalInteractionReason'}]},
      'venue': {'id': '4bb6b9446edc76b0d771311c',
       'name': "Wendy's",
       'location': {'crossStreet': 'Morningside & Sheppard',
        'lat': 43.80744841934756,
        'lng': -79.19905558052072,
        'labeledLatLngs': [{'label': 'display',
          'lat': 43.80744841934756,
          'lng': -79.19905558052072}],
        'distance': 387,
        'cc': 'CA',
        'city': 'Toronto',
        'state': 'ON',
        'country': 'Canada',
        'formattedAddress': ['Toronto ON', 'Canada']},
       'categories': [{'id': '4bf58dd8d48988d16e941735',
         'name': 'Fast Food Restaurant',
         'pluralName': 'Fast Food Restaurants',
         'shortName': 'Fast Food',
         'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/
fastfood_',
          'suffix': '.png'},
         'primary': True}],
       'photos': {'count': 0, 'groups': []}},
      'referralId': 'e-0-4bb6b9446edc76b0d771311c-0'},
     {'reasons': {'count': 0,
       'items': [{'summary': 'This spot is popular',
         'type': 'general',
         'reasonName': 'globalInteractionReason'}]},
      'venue': {'id': '5539e7d2498edaf4b02673ca',
       'name': 'Interprovincial Group',
       'location': {'address': '1315 Morningside Avenue',
        'lat': 43.8056297,
        'lng': -79.2003784,
        'labeledLatLngs': [{'label': 'display',
          'lat': 43.8056297,
          'lng': -79.2003784}],
        'distance': 498,
        'postalCode': 'M1B 3C5',
        'cc': 'CA',
        'city': 'Scarborough',
        'state': 'ON',
        'country': 'Canada',
        'formattedAddress': ['1315 Morningside Avenue',
         'Scarborough ON M1B 3C5',
         'Canada']},
       'categories': [{'id': '52f2ab2ebcbc57f1066b8b28',
         'name': 'Print Shop',
         'pluralName': 'Print Shops',
         'shortName': 'Print Shop',
         'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/shops
/default_',
```

In [38]:
```
# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

In [39]:
```
venues = results['response']['groups'][0]['items']

nearby_venues = json_normalize(venues) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.la
t', 'venue.location.lng']
nearby_venues =nearby_venues.loc[:, filtered_columns]

# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_typ
e, axis=1)

# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.col
umns]

nearby_venues.head()
```

Out[39]:

	name	categories	lat	lng
0	Wendy's	Fast Food Restaurant	43.807448	-79.199056
1	Interprovincial Group	Print Shop	43.805630	-79.200378

In [40]:
```
print('{} venues were returned by Foursquare.'.format(nearby_venues.shap
e[0]))
```

```
2 venues were returned by Foursquare.
```

2. Explore Neighborhoods in Toranto

In [41]:
```
#### Let's create a function to repeat the same process to all the neigh
borhoods in Toranto
```

In [42]:
```
def getNearbyVenues(names, latitudes, longitudes, radius=500):

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

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

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['ite
ms']

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

    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)
```

In [43]:
```
toranto_venues = getNearbyVenues(names=toranto_table['Neighborhood'],
                                 latitudes=toranto_table['Latitude'],
                                 longitudes=toranto_table['Longitude']
                                )


toranto_venues = getNearbyVenues(names=toranto_table['Neighbuorhood'],
                                 latitudes=toranto_table['Latitude'],
                                 longitudes=toranto_table['Longitude']
                                )
```

```
Rouge,Malvern
Port Union,Rouge Hill,Highland Creek
Guildwood,Morningside,West Hill
Woburn
Cedarbrae
Scarborough Village
East Birchmount Park,Ionview,Kennedy Park
Golden Mile,Oakridge,Clairlea
Cliffcrest,Scarborough Village West,Cliffside
Cliffside West,Birch Cliff
Wexford Heights,Dorset Park,Scarborough Town Centre
Maryvale,Wexford
Agincourt
Sullivan,Clarks Corners,Tam O'Shanter
Milliken,Agincourt North,L'Amoreaux East,Steeles East
L'Amoreaux West
Upper Rouge
Hillcrest Village
Fairview,Oriole,Henry Farm
Bayview Village
Silver Hills,York Mills
Willowdale,Newtonbrook
Willowdale South
York Mills West
Willowdale West
Parkwoods
Don Mills North
Don Mills South,Flemingdon Park
Wilson Heights,Downsview North,Bathurst Manor
Northwood Park,York University
Downsview East,CFB Toronto
Downsview West
Downsview Central
Downsview Northwest
Victoria Village
Woodbine Gardens,Parkview Hill
Woodbine Heights
The Beaches
Leaside
Thorncliffe Park
East Toronto
The Danforth West,Riverdale
The Beaches West,India Bazaar
Studio District
Lawrence Park
Davisville North
North Toronto West
Davisville
Moore Park,Summerhill East
South Hill,Rathnelly,Summerhill West,Deer Park,Forest Hill SE
Rosedale
Cabbagetown,St. James Town
Church and Wellesley
Regent Park,Harbourfront
Ryerson,Garden District
St. James Town
Berczy Park
Central Bay Street
King,Adelaide,Richmond
Toronto Islands,Harbourfront East,Union Station
Toronto Dominion Centre,Design Exchange
Commerce Court,Victoria Hotel
Lawrence Manor East,Bedford Park
Roselawn
Forest Hill North,Forest Hill West
The Annex,Yorkville,North Midtown
University of Toronto,Harbord
Kensington Market,Grange Park,Chinatown
```

```
---------------------------------------------------------------------------
--
KeyError                                  Traceback (most recent call las
t)
~/anaconda3/lib/python3.7/site-packages/pandas/core/indexes/base.py in ge
t_loc(self, key, method, tolerance)
   2656             try:
-> 2657                 return self._engine.get_loc(key)
   2658             except KeyError:

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjec
tHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjec
tHashTable.get_item()

KeyError: 'Neighbuorhood'

During handling of the above exception, another exception occurred:

KeyError                                  Traceback (most recent call las
t)
<ipython-input-43-be2cc7762c4c> in <module>
      5
      6
----> 7 toranto_venues = getNearbyVenues(names=toranto_table['Neighbuorho
od'],
      8                                   latitudes=toranto_table['Latit
ude'],
      9                                   longitudes=toranto_table['Long
itude']

~/anaconda3/lib/python3.7/site-packages/pandas/core/frame.py in __getitem
__(self, key)
   2925             if self.columns.nlevels > 1:
   2926                 return self._getitem_multilevel(key)
-> 2927             indexer = self.columns.get_loc(key)
   2928             if is_integer(indexer):
   2929                 indexer = [indexer]

~/anaconda3/lib/python3.7/site-packages/pandas/core/indexes/base.py in ge
t_loc(self, key, method, tolerance)
   2657                 return self._engine.get_loc(key)
   2658             except KeyError:
-> 2659                 return self._engine.get_loc(self._maybe_cast_inde
xer(key))
   2660         indexer = self.get_indexer([key], method=method, toleranc
e=tolerance)
   2661         if indexer.ndim > 1 or indexer.size > 1:

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjec
tHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjec
tHashTable.get_item()

KeyError: 'Neighbuorhood'
```

In [44]:
```
print(toranto_venues.shape)
toranto_venues.head()
```

```
(2243, 7)
```

Out[44]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Ven Categ
0	Rouge,Malvern	43.806686	-79.194353	Wendy's	43.807448	-79.199056	Fast Fo Restaur
1	Rouge,Malvern	43.806686	-79.194353	Interprovincial Group	43.805630	-79.200378	Print Sh
2	Port Union,Rouge Hill,Highland Creek	43.784535	-79.160497	Royal Canadian Legion	43.782533	-79.163085	E
3	Port Union,Rouge Hill,Highland Creek	43.784535	-79.160497	Affordable Toronto Movers	43.787919	-79.162977	Mov Tar
4	Guildwood,Morningside,West Hill	43.763573	-79.188711	Swiss Chalet Rotisserie & Grill	43.767697	-79.189914	Piz Pla

Let's check how many venues were returned for each neighborhood

In [45]:
```
toranto_venues.groupby('Neighborhood').count()
```

Out[45]:

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venu Categor
Agincourt	4	4	4	4	4	
Albion Gardens,Beaumond Heights,Humbergate,Jamestown,Mount Olive,South Steeles,Thistletown,Silverstone	9	9	9	9	9	
Bayview Village	4	4	4	4	4	
Berczy Park	55	55	55	55	55	5
Business Reply Mail Processing Centre 969 Eastern	18	18	18	18	18	1
Cabbagetown,St. James Town	45	45	45	45	45	4
Caledonia-Fairbanks	5	5	5	5	5	
Canada Post Gateway Processing Centre	11	11	11	11	11	1
Cedarbrae	7	7	7	7	7	
Central Bay Street	83	83	83	83	83	8
Christie	16	16	16	16	16	1
Church and Wellesley	85	85	85	85	85	8
Cliffcrest,Scarborough Village West,Cliffside	2	2	2	2	2	
Cliffside West,Birch Cliff	4	4	4	4	4	
Cloverdale,West Deane Park,Princess Gardens,Martin Grove,Islington	1	1	1	1	1	
Commerce Court,Victoria Hotel	100	100	100	100	100	10
Davisville	35	35	35	35	35	3
Davisville North	7	7	7	7	7	
Don Mills North	6	6	6	6	6	
Don Mills South,Flemingdon Park	23	23	23	23	23	2
Dovercourt Village,Dufferin	19	19	19	19	19	1
Downsview Central	3	3	3	3	3	
Downsview East,CFB Toronto	2	2	2	2	2	
Downsview Northwest	5	5	5	5	5	
Downsview West	5	5	5	5	5	
East Birchmount Park,Ionview,Kennedy Park	7	7	7	7	7	
East Toronto	6	6	6	6	6	
Exhibition Place,Brockton,Parkdale Village	23	23	23	23	23	2
Fairview,Oriole,Henry Farm	64	64	64	64	64	6
Forest Hill North,Forest Hill West	4	4	4	4	4	
Glencairn	5	5	5	5	5	
Golden Mile,Oakridge,Clairlea	7	7	7	7	7	
Guildwood,Morningside,West Hill	7	7	7	7	7	

Let's find out how many unique categories can be curated from all the returned venues

In [46]:
```
print('There are {} uniques categories.'.format(len(toranto_venues['Venue Category'].unique())))
```

```
There are 275 uniques categories.
```

3. Analyze Each Neighborhood

In [47]:
```
# one hot encoding
toranto_onehot = pd.get_dummies(toranto_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
toranto_onehot['Neighborhood'] = toranto_venues['Neighborhood']

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

toranto_onehot.head()
```

Out[47]:

	Yoga Studio	Accessories Store	Afghan Restaurant	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	American Restaurant	Ant
0	0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	

Next, let's group rows by neighborhood and by taking the mean of the frequency of occurrence of each category and then let us confirm the new size

In [48]:
```
toranto_grouped = toranto_onehot.groupby('Neighborhood').mean().reset_in
dex()
toranto_grouped
```

Out[48]:

	Neighborhood	Yoga Studio	Accessories Store	Afghan Restaurant	Airport	Airport Food Court	Airport Gate	Airport Lounge
0	Agincourt	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	Albion Gardens,Beaumond Heights,Humbergate,Jam...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	Bayview Village	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	Berczy Park	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	Business Reply Mail Processing Centre 969 Eastern	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
5	Cabbagetown,St. James Town	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
6	Caledonia-Fairbanks	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
7	Canada Post Gateway Processing Centre	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
8	Cedarbrae	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
9	Central Bay Street	0.012048	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10	Christie	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11	Church and Wellesley	0.011765	0.000000	0.011765	0.000000	0.000000	0.000000	0.000000
12	Cliffcrest,Scarborough Village West,Cliffside	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
13	Cliffside West,Birch Cliff	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
14	Cloverdale,West Deane Park,Princess Gardens,Ma...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
15	Commerce Court,Victoria Hotel	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
16	Davisville	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
17	Davisville North	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
18	Don Mills North	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
19	Don Mills South,Flemingdon Park	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
20	Dovercourt Village,Dufferin	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
21	Downsview Central	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
22	Downsview East,CFB Toronto	0.000000	0.000000	0.000000	0.500000	0.000000	0.000000	0.000000
23	Downsview Northwest	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
24	Downsview West	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25	East Birchmount Park,Ionview,Kennedy Park	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
26	East Toronto	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
27	Exhibition Place,Brockton,Parkdale Village	0.043478	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
28	Fairview,Oriole,Henry Farm	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
29	Forest Hill North,Forest Hill West	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
30	Glencairn	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

In [49]:
```
toranto_grouped.shape
```

Out[49]:
```
(97, 275)
```

Let's print each neighborhood along with the top 5 most common venues

In [50]:
```
num_top_venues = 5

for hood in toranto_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = toranto_grouped[toranto_grouped['Neighborhood'] == hood].T.re
set_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=Tru
e).head(num_top_venues))
    print('\n')
```

```
----Agincourt----
            venue  freq
0  Sandwich Place  0.25
1    Skating Rink  0.25
2  Breakfast Spot  0.25
3         Lounge  0.25
4   Movie Theater  0.00


----Albion Gardens,Beaumond Heights,Humbergate,Jamestown,Mount Olive,Sout
h Steeles,Thistletown,Silverstone----
                venue  freq
0        Grocery Store  0.22
1          Pizza Place  0.11
2  Fried Chicken Joint  0.11
3           Beer Store  0.11
4          Coffee Shop  0.11


----Bayview Village----
                venue  freq
0  Japanese Restaurant  0.25
1                Café  0.25
2                Bank  0.25
3   Chinese Restaurant  0.25
4         Music Store  0.00


----Berczy Park----
                venue  freq
0         Coffee Shop  0.11
1        Cocktail Bar  0.05
2         Cheese Shop  0.04
3      Farmers Market  0.04
4  Seafood Restaurant  0.04


----Business Reply Mail Processing Centre 969 Eastern----
                venue  freq
0  Light Rail Station  0.11
1          Comic Shop  0.06
2      Farmers Market  0.06
3             Brewery  0.06
4       Moving Target  0.06


----Cabbagetown,St. James Town----
                venue  freq
0         Coffee Shop  0.09
1                 Pub  0.04
2  Italian Restaurant  0.04
3                Park  0.04
4          Restaurant  0.04


----Caledonia-Fairbanks----
                  venue  freq
0         Women's Store   0.2
1  Fast Food Restaurant   0.2
2              Pharmacy   0.2
3                  Park   0.2
4                Market   0.2


----Canada Post Gateway Processing Centre----
                      venue  freq
0               Coffee Shop  0.18
1                     Hotel  0.18
```

Let's put that into a pandas dataframe

In [51]:
```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

Now let's create the new dataframe and display the top 10 venues for each neighborhood.

In [52]:
```
num_top_venues = 10

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

# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators
[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = toranto_grouped['Neighborh
ood']

for ind in np.arange(toranto_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venue
s(toranto_grouped.iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted.head()
```

Out[52]:

	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 Mos Commo Venu
0	Agincourt	Skating Rink	Breakfast Spot	Sandwich Place	Lounge	Event Space	Ethiopian Restaurant	Empanad Restauran
1	Albion Gardens,Beaumond Heights,Humbergate,Jam...	Grocery Store	Pizza Place	Fried Chicken Joint	Coffee Shop	Sandwich Place	Beer Store	Fast Foo Restauran
2	Bayview Village	Café	Japanese Restaurant	Bank	Chinese Restaurant	Women's Store	Dog Run	Done Restauran
3	Berczy Park	Coffee Shop	Cocktail Bar	Steakhouse	Farmers Market	Bakery	Beer Bar	Chees Sho
4	Business Reply Mail Processing Centre 969 Eastern	Light Rail Station	Comic Shop	Garden	Brewery	Skate Park	Farmers Market	Sp

4. Cluster Neighborhoods

Run k-means to cluster the neighborhood into 5 clusters.

In [53]:
```
# set number of clusters
kclusters = 5

toranto_grouped_clustering = toranto_grouped.drop('Neighborhood', 1)

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

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

Out[53]:
```
array([1, 1, 1, 1, 1, 1, 0, 1, 1, 1], dtype=int32)
```

In [54]:
```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

Let's create a new dataframe that includes the cluster as well as the top 10 venues for each neighborhood.

In [55]:
```
num_top_venues = 10

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

# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators
[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = toranto_grouped['Neighborh
ood']

for ind in np.arange(toranto_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venue
s(toranto_grouped.iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted.head()
```

Out[55]:

	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 Mos Commo Venu
0	Agincourt	Skating Rink	Breakfast Spot	Sandwich Place	Lounge	Event Space	Ethiopian Restaurant	Empanad Restauran
1	Albion Gardens,Beaumond Heights,Humbergate,Jam...	Grocery Store	Pizza Place	Fried Chicken Joint	Coffee Shop	Sandwich Place	Beer Store	Fast Foo Restauran
2	Bayview Village	Café	Japanese Restaurant	Bank	Chinese Restaurant	Women's Store	Dog Run	Done Restauran
3	Berczy Park	Coffee Shop	Cocktail Bar	Steakhouse	Farmers Market	Bakery	Beer Bar	Chees Sho
4	Business Reply Mail Processing Centre 969 Eastern	Light Rail Station	Comic Shop	Garden	Brewery	Skate Park	Farmers Market	Sp

4. Cluster Neighborhoods

Run k-means to cluster the neighborhood into 5 clusters.

In [56]:
```
# set number of clusters
kclusters = 5

toranto_grouped_clustering = toranto_grouped.drop('Neighborhood', 1)

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

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

Out[56]:
```
array([1, 1, 1, 1, 1, 1, 0, 1, 1, 1], dtype=int32)
```

Let's create a new dataframe that includes the cluster as well as the top 10 venues for each neighborhood.

In [57]:
```
# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

toranto_merged = toranto_table

# merge toronto_grouped with toronto_table to add latitude/longitude for
each neighborhood
toranto_merged = toranto_merged.join(neighborhoods_venues_sorted.set_ind
ex('Neighborhood'), on='Neighborhood')

toranto_merged.head() # check the last columns!
```

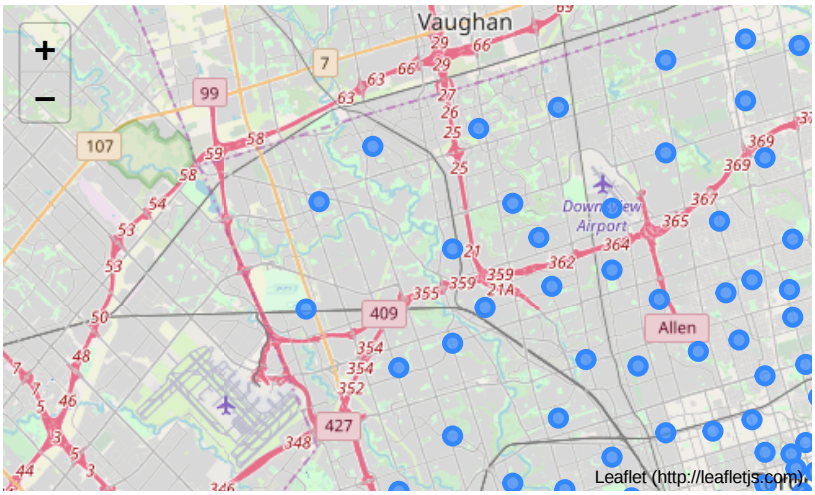
Out[57]:

	Postcode	Borough	Neighborhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Mo Commo Venu
0	M1B	Scarborough	Rouge,Malvern	43.806686	-79.194353	3.0	Fast Food Restaurant	Print Sh
1	M1C	Scarborough	Port Union,Rouge Hill,Highland Creek	43.784535	-79.160497	1.0	Moving Target	B
2	M1E	Scarborough	Guildwood,Morningside,West Hill	43.763573	-79.188711	1.0	Pizza Place	Electroni Sto
3	M1G	Scarborough	Woburn	43.770992	-79.216917	1.0	Coffee Shop	Korea Restaura
4	M1H	Scarborough	Cedarbrae	43.773136	-79.239476	1.0	Hakka Restaurant	Athletics Spor

Finally, let's visualize the resulting clusters

In [61]:
```
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(toranto_merged['Latitude'], toranto_me
rged['Longitude'], toranto_merged['Neighborhood'], toranto_merged['Clust
er Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_ht
ml=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,

        fill=True,

        fill_opacity=0.7).add_to(map_clusters)

map_clusters
#color=rainbow[cluster-1],
#fill_color=rainbow[cluster-1],
```

Out[61]:



5. Examine Clusters

Cluster 1

In [62]:
```
toranto_merged.loc[toranto_merged['Cluster Labels'] == 0, toranto_merge
d.columns[[1] + list(range(5, toranto_merged.shape[1]))]]
```

Out[62]:

	Borough	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 Cor
14	Scarborough	0.0	Playground	Park	Coffee Shop	Women's Store	Dumpling Restaurant	Discount Store	Div
23	North York	0.0	Park	Bank	Convenience Store	Women's Store	Eastern European Restaurant	Dog Run	Rest
25	North York	0.0	Fast Food Restaurant	Food & Drink Shop	Bus Stop	Park	Dive Bar	Dog Run	Rest
30	North York	0.0	Airport	Park	Women's Store	Eastern European Restaurant	Dive Bar	Dog Run	Rest
31	North York	0.0	Grocery Store	Shopping Mall	Park	Bank	Women's Store	Dive Bar	Do
40	East York	0.0	Park	Pizza Place	Coffee Shop	Convenience Store	Intersection	Drugstore	Dis
44	Central Toronto	0.0	Park	Bus Line	Construction & Landscaping	Swim School	Women's Store	Dumpling Restaurant	Do
48	Central Toronto	0.0	Playground	Park	Tennis Court	Women's Store	Donut Shop	Diner	Dis
50	Downtown Toronto	0.0	Park	Playground	Building	Trail	Women's Store	Drugstore	Dis
73	York	0.0	Field	Park	Hockey Arena	Trail	Eastern European Restaurant	Dive Bar	Do
74	York	0.0	Women's Store	Market	Park	Fast Food Restaurant	Pharmacy	Grocery Store	Eth Rest
79	North York	0.0	Basketball Court	Park	Construction & Landscaping	Bakery	Women's Store	Doner Restaurant	
90	Etobicoke	0.0	Park	Pool	River	Women's Store	Donut Shop	Diner	Dis
94	Etobicoke	0.0	Bank	Women's Store	Eastern European Restaurant	Dog Run	Doner Restaurant	Donut Shop	Dru
100	Etobicoke	0.0	Park	Bus Line	Mobile Phone Shop	Women's Store	Dumpling Restaurant	Dog Run	Rest

Cluster 2

In [63]:
```
toranto_merged.loc[toranto_merged['Cluster Labels'] == 1, toranto_merge
d.columns[[1] + list(range(5, toranto_merged.shape[1]))]]
```

Out[63]:

	Borough	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	Scarborough	1.0	Moving Target	Bar	Dumpling Restaurant	Dive Bar	Dog Run	Doner Restaurant
2	Scarborough	1.0	Pizza Place	Electronics Store	Intersection	Rental Car Location	Medical Center	Mexican Restaurant
3	Scarborough	1.0	Coffee Shop	Korean Restaurant	Women's Store	Dive Bar	Dog Run	Doner Restaurant
4	Scarborough	1.0	Hakka Restaurant	Athletics & Sports	Bakery	Caribbean Restaurant	Thai Restaurant	Bank
6	Scarborough	1.0	Discount Store	Bus Station	Convenience Store	Department Store	Coffee Shop	Chinese Restaurant
7	Scarborough	1.0	Bakery	Bus Line	Fast Food Restaurant	Soccer Field	Bus Station	Drugstore
8	Scarborough	1.0	Motel	American Restaurant	Women's Store	Diner	Dive Bar	Dog Run
9	Scarborough	1.0	College Stadium	Skating Rink	General Entertainment	Café	Empanada Restaurant	Electronics Store
10	Scarborough	1.0	Indian Restaurant	Pet Store	Vietnamese Restaurant	Light Rail Station	Latin American Restaurant	Chinese Restaurant
11	Scarborough	1.0	Breakfast Spot	Bakery	Smoke Shop	Middle Eastern Restaurant	Women's Store	Dumpling Restaurant
12	Scarborough	1.0	Skating Rink	Breakfast Spot	Sandwich Place	Lounge	Event Space	Ethiopian Restaurant
13	Scarborough	1.0	Pizza Place	Thai Restaurant	Noodle House	Chinese Restaurant	Fast Food Restaurant	Bank
15	Scarborough	1.0	Fast Food Restaurant	Chinese Restaurant	Pharmacy	Nail Salon	Electronics Store	Bubble Tea Shop
17	North York	1.0	Dog Run	Pool	Golf Course	Mediterranean Restaurant	Women's Store	Dumpling Restaurant
18	North York	1.0	Clothing Store	Coffee Shop	Fast Food Restaurant	Electronics Store	Restaurant	Bakery
19	North York	1.0	Café	Japanese Restaurant	Bank	Chinese Restaurant	Women's Store	Dog Run
22	North York	1.0	Ramen Restaurant	Coffee Shop	Sushi Restaurant	Pizza Place	Sandwich Place	Japanese Restaurant
24	North York	1.0	Pizza Place	Coffee Shop	Discount Store	Pharmacy	Concert Hall	Comfort Food Restaurant
26	North York	1.0	Japanese Restaurant	Basketball Court	Gym / Fitness Center	Caribbean Restaurant	Baseball Field	Café
27	North York	1.0	Gym	Asian Restaurant	Grocery Store	Beer Store	Coffee Shop	Fast Food Restaurant
28	North York	1.0	Coffee Shop	Pharmacy	Frozen Yogurt Shop	Shopping Mall	Bridal Shop	Fast Food Restaurant

Cluster 3

In [64]:
```
toranto_merged.loc[toranto_merged['Cluster Labels'] == 2, toranto_merge
d.columns[[1] + list(range(5, toranto_merged.shape[1]))]]
```

Out[64]:

	Borough	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 Com V
5	Scarborough	2.0	Playground	Dumpling Restaurant	Discount Store	Dive Bar	Dog Run	Doner Restaurant	Donut Shop	Drug

Cluster 4

In [65]:
```
toranto_merged.loc[toranto_merged['Cluster Labels'] == 3, toranto_merge
d.columns[[1] + list(range(5, toranto_merged.shape[1]))]]
```

Out[65]:

	Borough	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 M Comm Ve
0	Scarborough	3.0	Fast Food Restaurant	Print Shop	Drugstore	Diner	Discount Store	Dive Bar	Dog Run	Do Restau

Cluster 5

In [66]:
```
toranto_merged.loc[toranto_merged['Cluster Labels'] == 4, toranto_merge
d.columns[[1] + list(range(5, toranto_merged.shape[1]))]]
```

Out[66]:

	Borough	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 Mo Comm Ven
97	North York	4.0	Baseball Field	Women's Store	Eastern European Restaurant	Dog Run	Doner Restaurant	Donut Shop	Drugstore	Dumpl Restaur

In [ ]:
```
```