Categorizing Neighborhood Restaurant Data in Toronto

Alex Mason

2020/06/15

Table of Contents

Abstract

This paper utilizes Postal Code data and latitude/longitudinal data for the city of Toronto. This data is then used to analyze and categorize restaurants and venue data accordingly based on neighborhoods in Toronto. This project aims to leverage restaurant venue data from Foursquare's 'Places API' and a 'k-means clustering' algorithm the help identify 'restaurant profiles' for the Toronto City neighborhoods.

Introduction

Background

Restaurants have always been an appealing market and business venture for cities especially as tourist attractions and as a form of culture. Most people even dine out once or twice a week to try something new or for the restaurant becoming a place to socialize among peers and other individuals with similar food interests. Restaurants are a means to help an excellent array of cooks and staff members share their culture and exquisite cooking skills with their city and even the world. Restaurants also pose as nice alternative to cooking at home and getting a chance to enjoy a variety of foods you may not have heard of before either. Food is a form of expression and sometimes we want to better understand these expressions and see if they represent something greater than just the food cooked and prepared each and every day.

Problem

Cities are composed of a variety of types of restaurants from Chinese cuisine, to fast foods restaurants like McDonalds or Wendy's, to Italian restaurants with pizza that will make your mouth water simply by looking at it. These restaurants not only provide the food and culture needs to local citizens in the city, but also to tourists and food critiques from around the world. For bigger cities like Toronto for examples, Restaurants are usually spread across the city and in a variety of location to cater to different neighborhoods and communities but all to different regions and suburbs as well. This 'Restaurant Ecosystem' if you will, changes and grows over time. Knowing this information, we want to know preferences of different areas within the city limits as people are looking for good new food alternatives to try as well as what is preferred and what is disliked in regards to the restaurants and the dishes they serve. Also, with a large online food presence with companies like Yelp and Google Reviews that can post recommendations and crowd-sourced reviews for restaurants, their environments and the staff experiences are often viewed by others to see if restaurants are worth visiting or are best to be avoided.

This project will aim to quantify a 'Restaurant Profile' for neighborhoods in the major metropolitan city of Toronto, ON to be able to identify similar clusters and 'clumps' of similar restaurant scenes.

Stakeholders

There are a variety of individuals and parties that may be interested in a model that can quantify neighborhoods simply based on categories of restaurants in the vicinity. Most notably, this model would be able to help potential home renters and buyers who may prefer to live in neighborhoods where there is a large variety of restaurants or even a lot of one particular type of restaurant (i.e. Japanese restaurants). These individuals could also view nearby neighborhoods as well, as some neighborhoods and suburbs of Toronto may be more expensive than others. So, to these individuals, if they can find other restaurant clusters with similar likings to their food preferences within neighborhoods nearby or nearby suburbs, it may help their decision with home choices as well. This may also apply to people considering opening up a restaurant of their own. If someone is thinking of opening a local Jamaican restaurant, but to their discover finds that there are already a large number of Jamaican restaurants within that particular area or neighborhood. With this information to their disposal they be able to look at possible relocating to cut down on potential rival business or find it may be of interest to open another type of restaurant in that area instead. This would give these owners a competitive advantage and ensure they are launching a business with less direct competition.

Methodology

Data Sources

Toronto Postal Codes Data

The data I am using for this project is the list of postal codes in Toronto, Canada dataset where the data is imported as a dataframe with the postal code, borough (district) and associated neighborhood. The image to the right displays a sample of the dataset that was imported:

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

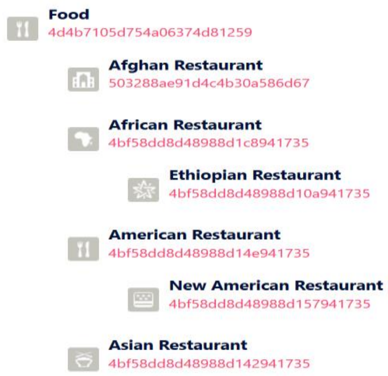
Geospatial Toronto Location Data Hosted by IBM

The geospatial data used for this project includes latitudinal, longitudinal and postal code data for the city of Toronto hosted by IBM. This data is primarily used to link up the postal code coordinates with the boroughs and neighborhood data in Toronto. This image to the right displays a sample of the dataset that was imported:

	Postal Code	Latitude	Longitude
0	M1B	43.806686	-79.194353
1	M1C	43.784535	-79.160497
2	M1E	43.763573	-79.188711
3	M1G	43.770992	-79.216917
4	M1H	43.773136	-79.239476

Foursquare – 'Places API'

For this project I am using Foursquare's 'Places API' to import some data related to food and restaurant venues under the food category. An important thing to note is that Foursquare identifies a 'venue' as a place that one can check into and is a registered establishment in this sense. Every one of Foursquare's 'venues' are assigned a 'categoryID' that relates back to the type of venue is it assigned with. The image to the right of this text provides some sample 'categoryIDs' that are provided by Foursquare:



Data Collection

Postal Code, Borough and, Neighborhood Name Data

The dataset is accessible through a Wikipedia page. The table from the webpage was acquired by utilizing a package titled bs4 with an imported subpackage called BeautifulSoup. BeautifulSoup allows us to pull the table right out of the Wikipedia and parse elements in the webpage and search directly for the table information we need. From there we search through the elements in the table and acquire all the data to be striped and placed in a custom Pandas dataframe inside of our Jupyter Notebook:

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

```
#utilize BeautifulSoup from bs4 to pull the data out
#of the wiki page and bring it into our notebook
Bsoup = BeautifulSoup(source, "html.parser")
table = Bsoup.find('table')
```

```
#This dataframe will consist of 3 columns: PostalCode, Borough, and Neig
column_Names = ['Postalcode', 'Borough', 'Neighborhood']
df = pd.DataFrame(columns = column_Names)
```

```
#Now we search for all postalcodes, boroughs, and neighborhoods!
for tr_cell in table.find_all('tr'): #table reader
    row_data=[]
    for td_cell in tr_cell.find_all('td'): #data reader
        row_data.append(td_cell.text.strip())
    if len(row_data)==3:
        df.loc[len(df)] = row_data
df.head()
```

Geospatial Toronto Location Data

Next, I simply import the Geospatial data for the city of Toronto to get the Latitude and Longitude coordinates for each postal code. The pandas function ‘read_csv’ is used to read in the data from the hosted .csv file from IBM:

```
geo_df = pd.read_csv('http://cocl.us/Geospatial_data')
geo_df.head()
```

Importing Foursquare Food-Related Venue Data

To be able to access the Foursquare ‘Venue Categories’ used to identify each type of venue, we have to submit a ‘get’ request. A ‘get’ request requires us to format a link to the url ‘https://api.foursquare.com/v2/venues/explore?’ as the endpoint, our login credentials for Foursquare, the version of the API we are calling, relevant Toronto latitude and longitude coordinates, the search radius for the venues, and the limit of how many venues we are requesting. The sample code below shows the variables and function that accesses the Foursquare API to access the necessary information:

```
# set variables for login credentials for Foursquare
CLIENT_ID = 'SUSG0YCSQ5LNWEU2ANEJYNI5MTTCCB4VMQRFTDSFWRXH0N5I'
CLIENT_SECRET = '0DGE5GS5KY42YD34DYQHOCZMS14MMTW10FWHYJ04QTMJIOBO'
VERSION = '20200606'
```

```
def getNearbyVenues(names, latitudes, longitudes, radius=1000, ID=''):
    LIMIT=5000
    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_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        if(ID!=''):
            url=url+'&ID={}'
            url=url.format(ID)

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

        # 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)
```

```
TO_venues = getNearbyVenues(names=geo_data['Neighborhood'],
                                   latitudes=geo_data['Latitude'],
                                   longitudes=geo_data['Longitude'],
                                   radius=1000,
                                   ID='4d4b7105d754a06374d81259'
                                  )
```

```
Rouge, Malvern
Highland Creek, Rouge Hill, Port Union
Guildwood, Morningside, West Hill
Woburn
Cedarbrae
Scarborough Village
East Birchmount Park, Ionview, Kennedy Park
Clairlea, Golden Mile, Oakridge
Cliffcrest, Cliffside, Scarborough Village West
Birch Cliff, Cliffside West
Dorset Park, Scarborough Town Centre, Wexford Heights
Maryvale, Wexford
Agincourt
Clarks Corners, Sullivan, Tam O'Shanter
Agincourt North, L'Amoreaux East, Milliken, Steeles East
```

When we load this data into a dataframe. We get the following:

```
TO_venues.head()
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Malvern, Rouge	43.806686	-79.194353	Harvey's	43.800020	-79.198307	Restaurant
1	Malvern, Rouge	43.806686	-79.194353	Wendy's	43.802008	-79.198080	Fast Food Restaurant
2	Malvern, Rouge	43.806686	-79.194353	Wendy's	43.807448	-79.199056	Fast Food Restaurant
3	Malvern, Rouge	43.806686	-79.194353	RBC Royal Bank	43.798782	-79.197090	Bank
4	Malvern, Rouge	43.806686	-79.194353	Caribbean Wave	43.798558	-79.195777	Caribbean Restaurant

Data Cleansing (Source Data)

The initial dataset that was imported for the Toronto Postal Codes actually had some columns where certain neighborhoods and boroughs were not properly assigned to the associated Postal Code. As you can see to the right:

	Postalcode	Borough	Neighborhood
0	M1A	Not assigned	Not assigned
1	M2A	Not assigned	Not assigned
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Harbourfront

To fix this, we get rid of the columns with values 'Not Assigned' so we are just displaying the properly assigned neighborhoods. The below code displays how this is done:

```
df = df[df.Borough != 'Not assigned']
# we the combine the neighborhoods with the same Postalcode
df1 = df.groupby(['Postalcode', 'Borough'], sort=False).agg(', '.join)
df1.reset_index(inplace = True)
# now replace name of neighborhoods with "not assigned" rows with names of Borough
df1['Neighborhood'] = np.where(df1['Neighborhood'] == 'Not assigned',df1['Borough'],
df1['Neighborhood'])
df1.head()
```

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

Next, we merge our two dataframes; the one for the neighborhoods and boroughs associated with the postal codes; and the one with the longitudinal and latitudinal coordinates. The result is the follow set of code with the following merged dataframe:

```
geo_df.rename(columns={'Postal Code':'Postalcode'},inplace=True)
geo_merged = pd.merge(geo_df, df1, on='Postalcode')
```

```
geo_data = geo_merged[['Postalcode', 'Borough', 'Neighborhood', 'Latitude', 'Longitude']]
geo_data.head()
```

	Postalcode	Borough	Neighborhood	Latitude	Longitude
0	M1B	Scarborough	Rouge, Malvern	43.806686	-79.194353
1	M1C	Scarborough	Highland Creek, Rouge Hill, Port Union	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

Data Analysis

The data sets and images captured below capture the processes I used for exploring the dataset that was received from Foursquare to better understand the venue categories and what venues exactly were in my dataset.

I started by checking which venue categories were captured in the dataframe by grouping each entry by distinct entry by ‘Venue Category’ and displaying the results:

```
unique_C = len(TO_df['Venue Category'].unique())
print(f'There are {unique_C} unique venue categories in this dataframe')
TO_df.groupby('Venue Category')['Venue Category'].count().sort_values(ascending=False)
```

```
There are 331 unique venue categories in this dataframe

Venue Category
Coffee Shop            378
Café                   210
Park                   152
Pizza Place            151
Restaurant             131
                      ...
Lighting Store           1
Mac & Cheese Joint       1
Market                   1
Massage Studio           1
Housing Development      1
Name: Venue Category, Length: 331, dtype: int64
```

Next, we check to verify that there are no null values in the dataset:

```
TO_df.isnull().values.any()
```

```
False
```

Finally, I check how many unique different venues there in the dataframe. What this means is restaurants with unique names where names are not duplicated in the dataframe:

```
unique_V = len(TO_df['Venue'].unique())
print(f'There are {unique_V} unique venues in the dataframe')
```

```
There are 2800 unique venues in the dataframe
```

Data Pre-Processing

Additional Data Cleansing

The preliminary dataset was cleansed in relation to the data analysis section above. To better organize the data and get rid of unnecessary entries, we remove the entries that are no associated with the 'venue category' for restaurants. We then display the new recognized dataframe which can be viewed below:

```
TO_df = TO_df.rename(columns={'Venue Category': 'Venue_Category'})
TO_df = TO_df[TO_df.Venue_Category.str.contains('Restaurant',case=False)]
TO_df.head(10)
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue_Category
0	Malvern, Rouge	43.806686	-79.194353	Harvey's	43.800020	-79.198307	Restaurant
1	Malvern, Rouge	43.806686	-79.194353	Wendy's	43.802008	-79.198080	Fast Food Restaurant
2	Malvern, Rouge	43.806686	-79.194353	Wendy's	43.807448	-79.199056	Fast Food Restaurant
4	Malvern, Rouge	43.806686	-79.194353	Caribbean Wave	43.798558	-79.195777	Caribbean Restaurant
10	Malvern, Rouge	43.806686	-79.194353	Mr Jerk	43.801262	-79.199758	African Restaurant
13	Malvern, Rouge	43.806686	-79.194353	Charley's Exotic Cuisine	43.800982	-79.200233	Chinese Restaurant
16	Malvern, Rouge	43.806686	-79.194353	Swiss Chalet	43.800236	-79.198366	Restaurant
18	Rouge Hill, Port Union, Highland Creek	43.784535	-79.160497	Fratelli Village Pizzeria	43.784008	-79.169787	Italian Restaurant
27	Guildwood, Morningside, West Hill	43.763573	-79.188711	Swiss Chalet	43.768122	-79.190493	Restaurant
32	Guildwood, Morningside, West Hill	43.763573	-79.188711	KFC	43.768900	-79.185600	Fast Food Restaurant

We then check how many restaurant entries are left in the data frame, and how many unique restaurant venues are left in the dataframe:

```
Entries = TO_df.shape[0]
print(f'There are {Entries} entries left in the TO_df dataframe')
```

```
There are 1173 entries left in the TO_df dataframe
```

```
unique_V = len(TO_df['Venue'].unique())
print(f'There are {unique_V} unique venues in the dataframe')
```

```
There are 783 unique venues in the dataframe
```

```
unique_C = len(TO_df['Venue_Category'].unique())
print(f'There are {unique_C} unique venue categories in this dataframe')
TO_df.groupby('Venue_Category')['Venue_Category'].count().sort_values(ascending=False)
```

```
There are 61 unique venue categories in this dataframe

Venue_Category
Restaurant                  131
Italian Restaurant          108
Japanese Restaurant          88
Sushi Restaurant             78
Fast Food Restaurant         63
                           ...
Cajun / Creole Restaurant     1
Hawaiian Restaurant           1
Hotpot Restaurant             1
North Indian Restaurant       1
African Restaurant            1
Name: Venue_Category, Length: 61, dtype: int64
```

Now we can observe that we were successfully able to cut down our dataset a fair amount. We now observe that we only have 61 unique venues of interest, down from 331 and 783 unique venues down from 2800.

One Hot-Encoding Restaurant Categories

In order to use Foursquare's different category values and information to find similar neighborhoods that are comparable with restaurant venues and categories, we must create a one-hot-encoding representation of each entry. To do this, we use Pandas' 'get_dummies' function to numerate our values in our dataset to zeros and ones. This will help us assign a numerable value to each column and row value to determine if that particular restaurant category is in its assigned neighborhood row. Basically, 0 means no, 1 means yes. The following code below displays the output of our one hot-encoded variables as well as the shape our of dataframe:

```
TO_df_onehot = pd.get_dummies(TO_df[['Venue_Category']], prefix="", prefix_sep="")

TO_df_onehot['Neighborhood'] = TO_df['Neighborhood']

fixed_columns = [TO_df_onehot.columns[-1]] + list(TO_df_onehot.columns[:-1])
TO_df_onehot = TO_df_onehot[fixed_columns]

print("Shape of Dataframe:", TO_df_onehot.shape)
TO_df_onehot.head(10)
```

```
Shape of Dataframe: (1173, 62)
```

	Neighborhood	Afghan Restaurant	African Restaurant	American Restaurant	Asian Restaurant	Belgian Restaurant	Brazilian Restaurant	Cajun / Creole Restaurant	Cantonese Restaurant	Caribbean Restaurant	...	Sushi Restaurant
0	Malvern, Rouge	0	0	0	0	0	0	0	0	0	...	0
1	Malvern, Rouge	0	0	0	0	0	0	0	0	0	...	0
2	Malvern, Rouge	0	0	0	0	0	0	0	0	0	...	0
4	Malvern, Rouge	0	0	0	0	0	0	0	0	1	...	0
10	Malvern, Rouge	0	1	0	0	0	0	0	0	0	...	0
13	Malvern, Rouge	0	0	0	0	0	0	0	0	0	...	0
16	Malvern, Rouge	0	0	0	0	0	0	0	0	0	...	0
18	Rouge Hill, Port Union, Highland Creek	0	0	0	0	0	0	0	0	0	...	0
27	Guildwood, Morningside, West Hill	0	0	0	0	0	0	0	0	0	...	0
32	Guildwood, Morningside, West Hill	0	0	0	0	0	0	0	0	0	...	0

10 rows × 62 columns

We can also determine the amount of venues in each category of each neighborhood by summing the total index for each neighborhood shown below:

```
venue_count = TO_df_onehot.groupby('Neighborhood').sum().reset_index('Neighborhood')

venue_count.head(10)
```

	Neighborhood	Afghan Restaurant	African Restaurant	American Restaurant	Asian Restaurant	Belgian Restaurant	Brazilian Restaurant	Cajun / Creole Restaurant	Cantonese Restaurant	Caribbean Restaurant	...	Sushi Restaurant	Syrian Restaurant
0	Agincourt	0	0	0	0	0	0	0	1	2	...	1	0
1	Alderwood, Long Branch	0	0	0	0	0	0	0	0	0	...	0	0
2	Bathurst Manor, Wilson Heights, Downsview North	0	0	0	0	0	0	0	0	0	...	1	0
3	Bayview Village	0	0	0	0	0	0	0	0	0	...	0	0
4	Bedford Park, Lawrence Manor East	0	0	1	0	0	0	0	0	0	...	1	0
5	Berczy Park	0	0	1	0	0	0	0	0	0	...	0	0
6	Birch Cliff, Cliffside West	0	0	0	0	0	0	0	0	0	...	0	0
7	Brockton, Parkdale Village, Exhibition Place	0	0	1	0	0	0	0	0	1	...	0	0
8	Business reply mail Processing Centre, South C...	0	0	1	0	0	0	0	0	0	...	2	0

Data Visualization

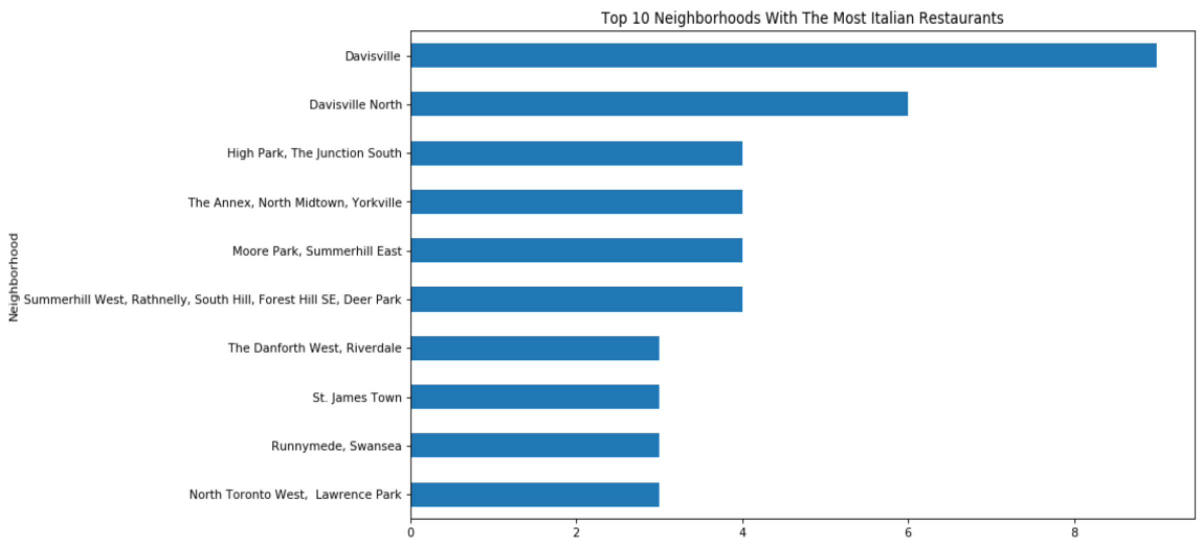
Using the dataframe of venue counts that was shown above, we can determine what the most popular restaurant venue categories. From there, we can display the top 10 neighborhoods by postal code with these top categories for restaurants in Toronto. The code and screenshots for this are displayed below:

```python
top_category_Neighborhoods = TO_df['Venue_Category'].value_counts()[:6].index.tolist()
top_category_Neighborhoods
```

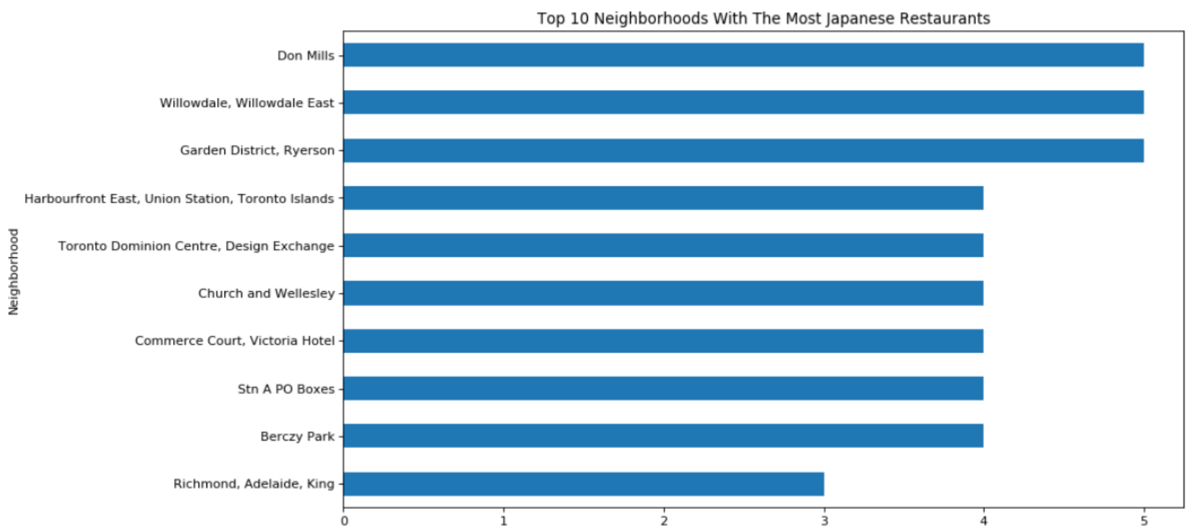
```
['Restaurant',
 'Italian Restaurant',
 'Japanese Restaurant',
 'Sushi Restaurant',
 'Fast Food Restaurant',
 'Thai Restaurant']
```

```python
top5_Categories = ['Italian Restaurant', 'Japanese Restaurant', 'Sushi Restaurant',
                   'Fast Food Restaurant', 'Thai Restaurant']
n = 10 # number of neighborhoods
for category in top5_Categories:
    plt.figure(num=None, figsize=(12, 7), dpi = 80, facecolor='w', edgecolor='k')
    plt.title(f'Top {n} Neighborhoods With The Most {category}s')
    get_top_Neighborhoods = venue_count[category].sort_values(ascending=False)[0:n]
    get_top_Neighborhoods = get_top_Neighborhoods.sort_values(ascending=True)

    get_top_Neighborhoods.plot.barh(y=category, rot=0)
```

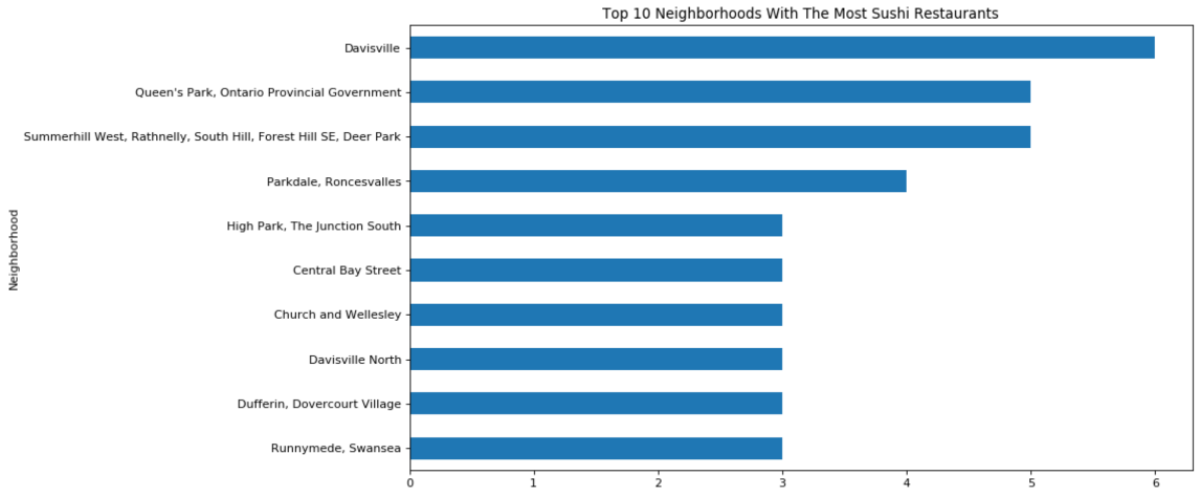
Top 10 Neighborhoods with Italian Restaurants



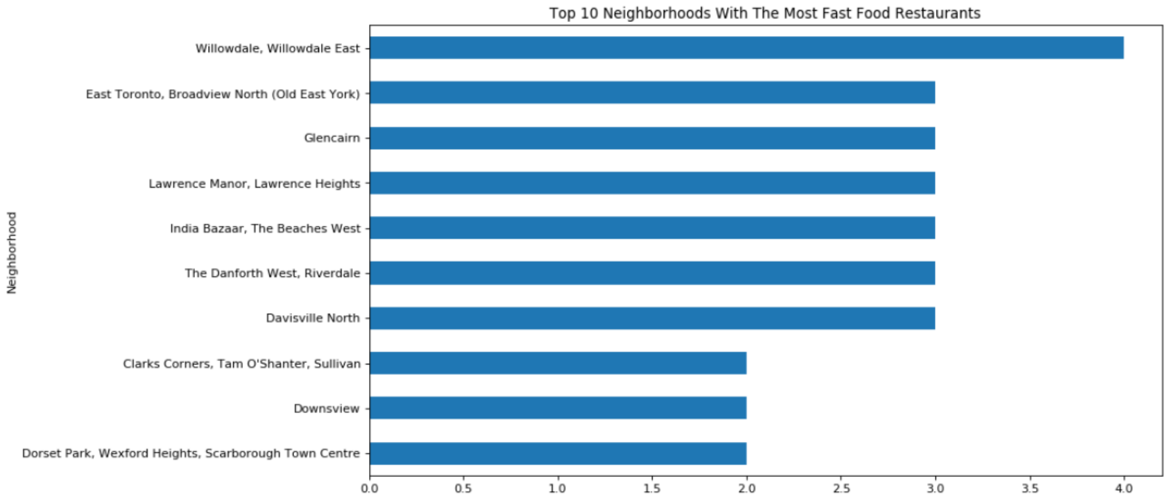
Top 10 Neighborhoods with Japanese Restaurants



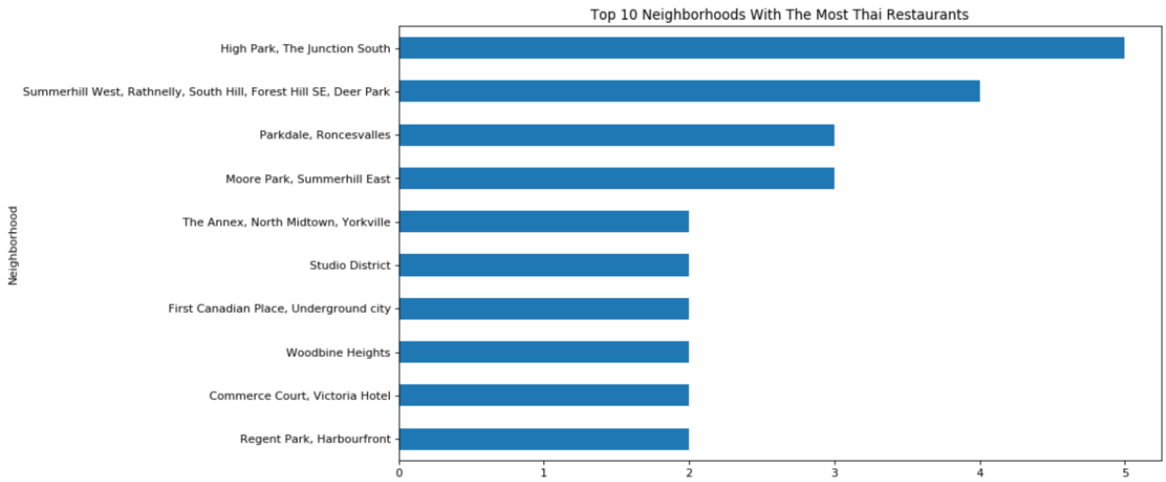
Top 10 Neighborhoods with Sushi Restaurants



Top 10 Neighborhoods with Fast Food Restaurants



Top 10 Neighborhoods with Thai Restaurants



Top Venue Categories Profile

The encoded dataset and restaurant-related category venues in Toronto were then used to quantify a restaurant profile for each neighborhood. For each venue category specified we determine the average distribution in each neighborhood. This information will then be used to fit a K-Means clustering approach to the data to help determine neighborhoods of similar restaurant venue profiles and completely distinct ones. We start by determining the total number of venues for each category as shown below:

```
venue_total = {}
for category in TO_df['Venue_Category']:
    venue_total[category] = venue_count[category].sum()

venue_total
```

```
{'Restaurant': 131,
 'Fast Food Restaurant': 63,
 'Caribbean Restaurant': 27,
 'African Restaurant': 1,
 'Chinese Restaurant': 47,
 'Italian Restaurant': 108,
 'Greek Restaurant': 35,
 'Indian Restaurant': 46,
 'Hakka Restaurant': 1,
 'Thai Restaurant': 53,
 'Japanese Restaurant': 88,
 'Asian Restaurant': 26,
 'Mexican Restaurant': 38,
 'Vietnamese Restaurant': 31,
 'American Restaurant': 42,
 'Korean Restaurant': 28,
 'Middle Eastern Restaurant': 40,
 'Seafood Restaurant': 32,
 'Indian Chinese Restaurant': 2,
 'Sri Lankan Restaurant': 1,
 'Cantonese Restaurant': 4,
 'Malay Restaurant': 3,
 'Sushi Restaurant': 78,
 'Latin American Restaurant': 8,
 'Mediterranean Restaurant': 17,
 'Filipino Restaurant': 3,
 'Taiwanese Restaurant': 2,
 'Vegetarian / Vegan Restaurant': 46,
 'Hotpot Restaurant': 1,
 'Ramen Restaurant': 26,
 'Indonesian Restaurant': 2,
 'Dumpling Restaurant': 3,
 'French Restaurant': 27,
 'Eastern European Restaurant': 8,
 'New American Restaurant': 12,
 'Dim Sum Restaurant': 4,
 'Doner Restaurant': 3,
 'Turkish Restaurant': 6,
 'Falafel Restaurant': 11,
 'Portuguese Restaurant': 5,
 'Afghan Restaurant': 2,
 'Ethiopian Restaurant': 8,
```

```
 'Tapas Restaurant': 6,
 'Cuban Restaurant': 3,
 'Pakistani Restaurant': 2,
 'Comfort Food Restaurant': 9,
 'Syrian Restaurant': 1,
 'German Restaurant': 5,
 'Modern European Restaurant': 6,
 'Theme Restaurant': 2,
 'Brazilian Restaurant': 4,
 'Persian Restaurant': 3,
 'Jewish Restaurant': 2,
 'Belgian Restaurant': 2,
 'Israeli Restaurant': 1,
 'South American Restaurant': 1,
 'Hawaiian Restaurant': 1,
 'Tibetan Restaurant': 3,
 'North Indian Restaurant': 1,
 'Cajun / Creole Restaurant': 1,
 'Moroccan Restaurant': 1}
```

We then create a new pandas dataframe to store our mean venue values which can be viewed in the code below:

```
venue_mean = pd.DataFrame()
for category, total in venue_total.items():
    venue_mean[category] = venue_count[category].apply(lambda x:x / total)

venue_mean.insert(0, 'Neighborhood', 'null')
venue_mean['Neighborhood'] = venue_count['Neighborhood']
venue_mean.head(5)
```

	Neighborhood	Restaurant	Fast Food Restaurant	Caribbean Restaurant	African Restaurant	Chinese Restaurant	Italian Restaurant	Greek Restaurant	Indian Restaurant	Hakka Restaurant	...	Persian Restaurant	Jewish Restaurant
0	Agincourt	0.007634	0.000000	0.074074	0.0	0.127660	0.000000	0.000000	0.021739	0.0	...	0.0	0.0
1	Alderwood, Long Branch	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.0	...	0.0	0.0
2	Bathurst Manor, Wilson Heights, Downsview North	0.007634	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.0	...	0.0	0.0
3	Bayview Village	0.007634	0.000000	0.000000	0.0	0.021277	0.000000	0.000000	0.000000	0.0	...	0.0	0.0
4	Bedford Park, Lawrence Manor East	0.015267	0.015873	0.000000	0.0	0.000000	0.027778	0.028571	0.021739	0.0	...	0.0	0.0

5 rows × 62 columns

Following this, we will create a function and a dataframe that will how us the top 5 restaurant categories for each neighborhood. The following code and output for the function can be viewed below:

```
def get_top_venue_Categories(row, num):
    row_categories = row.iloc[1:]
    row_categories_Sorted = row_categories.sort_values(ascending=False)

    return row_categories_Sorted.index.values[0:num] #returns sorted df
```

```python
num_of_TVenues = 5   # 5 top venues

indicators = ['st', 'nd', 'rd'] # nifty little trick

#sort columns by neighborhood top venues
columns = ['Neighborhood']
for indic in np.arange(num_of_TVenues):
    try:
        columns.append('{}{} Top Venue Category'.format(indic+1, indicators[indic]))
    except:
        columns.append('{}th Top Venue Category'.format(indic+1))

top_venue_categories = pd.DataFrame(columns=columns)
top_venue_categories['Neighborhood'] = venue_mean['Neighborhood']

for ind in np.arange(venue_mean.shape[0]):
    top_venue_categories.iloc[ind, 1:] = get_top_venue_Categories(venue_mean.iloc[ind, :], num_of_TVenues)

top_venue_categories.head(5)
```

	Neighborhood	1st Top Venue Category	2nd Top Venue Category	3rd Top Venue Category	4th Top Venue Category	5th Top Venue Category
0	Agincourt	Sri Lankan Restaurant	Malay Restaurant	Filipino Restaurant	Cantonese Restaurant	Chinese Restaurant
1	Alderwood, Long Branch	Moroccan Restaurant	Ramen Restaurant	Vegetarian / Vegan Restaurant	Taiwanese Restaurant	Filipino Restaurant
2	Bathurst Manor, Wilson Heights, Downsview North	Mediterranean Restaurant	Middle Eastern Restaurant	Sushi Restaurant	Restaurant	Ramen Restaurant
3	Bayview Village	Japanese Restaurant	Chinese Restaurant	Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant
4	Bedford Park, Lawrence Manor East	Comfort Food Restaurant	Greek Restaurant	Italian Restaurant	American Restaurant	Indian Restaurant

Results

KMeans Cluster Model

In the Scikit-learn library, there is a K-Means clustering function that we use to determine similar neighborhoods in the city of Toronto based on their restaurant venue category mean averages. The image below shows the dataframe we created being scaled into a K-Means model. This can then be visualized as an array of all the neighborhood clusters:

```python
kclusters = 15 # num of clusters

venue_grouping = venue_mean.drop('Neighborhood', 1)
kmeans = KMeans(n_clusters = kclusters, random_state=0).fit(venue_grouping)
 #display clusters in array
kmeans.labels_
```

```
array([ 8, 11,  4,  4,  4,  4,  4,  1,  4,  4,  4,  9,  4,  2,  4,  4,  4,
        3,  4,  4,  4,  4,  4,  4,  4,  4,  4,  4,  4,  4,  4,  4,  4,  6,
        4,  4, 12, 13,  4, 14,  4,  4,  4,  0,  5,  4,  4,  4,  4,  4,  4,
        4,  4,  4,  4,  4,  4, 13,  4,  4,  4,  4,  4,  4,  4,  4,  4,  4,
        7,  4,  4,  4,  4,  4,  0,  4, 10,  4, 14,  4,  4,  4,  4,  4,  4,
        4,  4,  4,  4,  4])
```

Now that we can visualize our clusters for neighborhoods in the form of an array, we can now create a new dataframe to combine everything together. We merge the neighborhood location data, cluster data, and the top venue category profile:

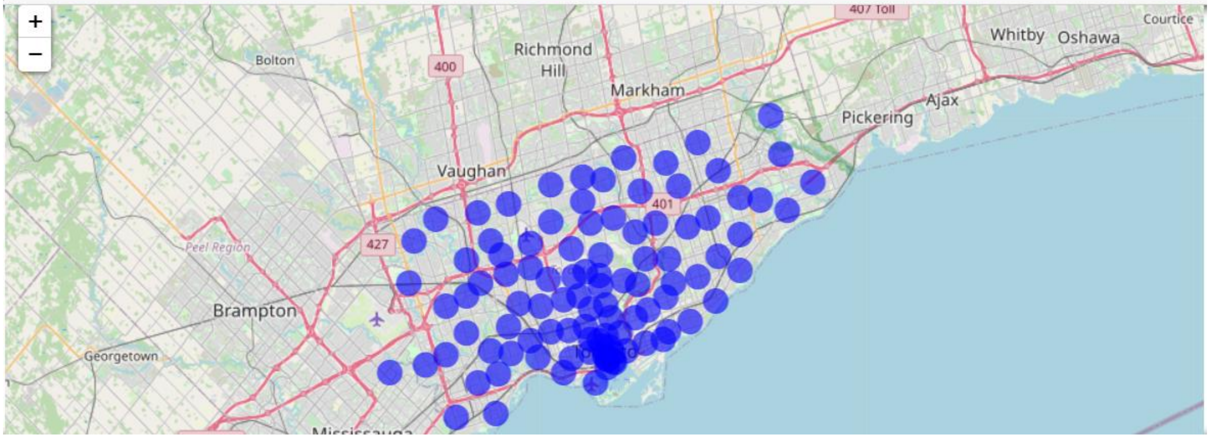
```
top_venue_categories.insert(1,'Cluster Labels', kmeans.labels_)
cluster_Df = geo_data.drop(columns=['Postalcode', 'Borough'])
# merge toronto data with previous dataframe with lat/lng for each neighborhood

cluster_Df = cluster_Df.join(top_venue_categories.set_index('Neighborhood'), on ='Neighborhood').dropna(axis=0)
cluster_Df.reset_index(drop=True, inplace=True)
cluster_Df['Cluster Labels'] = cluster_Df['Cluster Labels'].fillna("0").astype(int)
cluster_Df['Cluster Labels'] = cluster_Df['Cluster Labels'].astype('int64')

cluster_Df.head()
```

	Neighborhood	Latitude	Longitude	Cluster Labels	1st Top Venue Category	2nd Top Venue Category	3rd Top Venue Category	4th Top Venue Category	5th Top Venue Category
0	Malvern, Rouge	43.806686	-79.194353	5	African Restaurant	Caribbean Restaurant	Fast Food Restaurant	Chinese Restaurant	Restaurant
1	Rouge Hill, Port Union, Highland Creek	43.784535	-79.160497	4	Italian Restaurant	Moroccan Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant	Taiwanese Restaurant
2	Guildwood, Morningside, West Hill	43.763573	-79.188711	4	Fast Food Restaurant	Greek Restaurant	Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant
3	Woburn	43.770992	-79.216917	4	Indian Restaurant	Chinese Restaurant	Fast Food Restaurant	Moroccan Restaurant	Middle Eastern Restaurant
4	Cedarbrae	43.773136	-79.239476	9	Hakka Restaurant	Indian Restaurant	Caribbean Restaurant	Chinese Restaurant	Thai Restaurant

Cluster Visualization Map

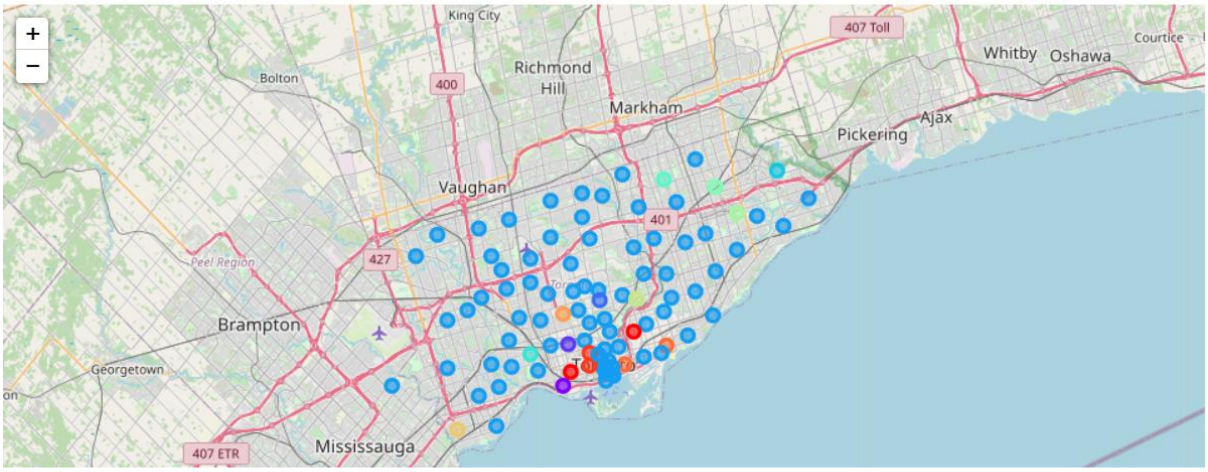
The following code used a python library folium to help us create a map of Toronto with bubble markers to visualize each of the different neighborhoods we selected:

```
toronto_map=folium.Map(location=[latitude,longitude],zoom_start=10)
# This is the map for the entire city, including the areas we excluded for only specific restaurants
for lat,long,borough,neighbourhood in zip(geo_data['Latitude'],geo_data['Longitude'],geo_data['Borough'],
                                          geo_data['Neighborhood']):
                label='{},{}'.format(neighbourhood,borough)
                label=folium.Popup(label,parse_html=True)
                folium.CircleMarker(
                [lat,long],
                radius=5,
                popup=label,
                color='',
                fill=True,
                fill_color='blue',
                fill_opacity=0.6,
                parse_html=True
                ).add_to(toronto_map)
display(toronto_map)
```



The next map we create is a visualization of all the clusters we created using our neighborhood restaurant category. Neighborhoods with similar color shades and patterns represent neighborhoods with similar restaurant profiles:

```
folium_map_TO = folium.Map(location=[latitude, longitude], zoom_start=10)

# setup map colors
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]

#markers for map
markers_colors = []
for lat, lon, poi, cluster in zip(cluster_Df['Latitude'], cluster_Df['Longitude'], cluster_Df['Neighborhood'],
cluster_Df['Cluster Labels']):
    label = folium.Popup(str(poi) + 'Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker([lat, lon],
                        radius = 5,
                        popup = label,
                        color = rainbow[cluster-1],
                        fill = True,
                        fill_color = rainbow[cluster-1],
                        fill_opacity=0.6).add_to(folium_map_TO)

display(folium_map_TO)
```



Cluster End Results

The full dataframe of all the resulting clusters is available to be viewed in the clusters section in the Appendix of this paper. Each cluster shows a list of neighborhoods with their respective top venue categories. Similarly, we can compare the resulting clusters to the original bar graph plots we initially displayed in the Data Visualization section of this paper. What is very fascinating about the K-Means cluster we created is that some of the clusters are very small sometimes only consisting of a few neighborhoods but most notably, the 4th cluster is very similar amongst many neighborhoods across the city. Where a majority of their top 5 restaurant categories are very closely related. This means that there are very similar interests in preference for food restaurants across the city of Toronto which is actually really cool. Regardless of the diversification that makes up the beautiful city of Toronto, there are still very similar preferences in terms of food preferences and restaurant preferences.

Conclusion

There are a variety of ways that machine learning and K-Means clustering algorithms can be applied to multi-dimensional datasets to help determine patterns and similarities within the contents of the data. The clustering of neighborhoods solely based on similarities surrounding food restaurant profiles (or any profile for that matter) can be generated with the help of high-quality venue location data. Finding high-quality location data can be quite a challenge as location data analysis models are only as good as the location data accessible for the models. Luckily, with the help of Foursquare's 'Places API' services, we can leverage this type of high-quality location data to our advantage in order to create outstanding data analysis models and visualization tools.

There are a number of ways this project can be expanded in the future. Foursquare's API could be further integrated and updated to determine more restaurants and 'restaurant-related' venue categories in the city of Toronto. With new information and venues that are added, new datasets from Foursquare could be acquired and merged to help create better cluster patterns with less clutter and more accurate data within our models. The next step in terms of clustering algorithms could be incorporating the DBSCAN clustering algorithm. The DBSCAN algorithm is better at maintaining rather dense clustering and better capability of ignoring unwanted outliers. We could then compare this method to the K-Means clustering algorithm to see the major/minor differences and consider which model creates a better approach to venue data. The clustering models could then become a basis for a recommendation system aimed at helping provide local restaurant owners and potential restaurant owner with a better picture of peoples preferences within specified neighborhoods and potentially their business' neighborhood.

References

1. Wikipedia (2020), "List of postal code of Canada: M", Accessed June 8th, 2020, https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M
2. Foursquare (2020), "Foursquare Venue Categories", Accessed June 8th, 2020, https://developer.foursquare.com/docs/build-with-foursquare/categories/
3. IBM (2020), "IBM Geospatial Toronto Location Data", Accessed June 8th, 2020, http://cocl.us/Geospatial_data

Appendix

Clusters

	Neighborhood	Latitude	Longitude	Cluster Labels	1st Top Venue Category	2nd Top Venue Category	3rd Top Venue Category	4th Top Venue Category	5th Top Venue Category
0	Malvern, Rouge	43.806686	-79.194353	5	African Restaurant	Caribbean Restaurant	Fast Food Restaurant	Chinese Restaurant	Restaurant
1	Rouge Hill, Port Union, Highland Creek	43.784535	-79.160497	4	Italian Restaurant	Moroccan Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant	Taiwanese Restaurant
2	Guildwood, Morningside, West Hill	43.763573	-79.188711	4	Fast Food Restaurant	Greek Restaurant	Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant
3	Woburn	43.770992	-79.216917	4	Indian Restaurant	Chinese Restaurant	Fast Food Restaurant	Moroccan Restaurant	Middle Eastern Restaurant
4	Cedarbrae	43.773136	-79.239476	9	Hakka Restaurant	Indian Restaurant	Caribbean Restaurant	Chinese Restaurant	Thai Restaurant
5	Scarborough Village	43.744734	-79.239476	4	Fast Food Restaurant	Japanese Restaurant	Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant
6	Kennedy Park, Ionview, East Birchmount Park	43.727929	-79.262029	4	Chinese Restaurant	Asian Restaurant	Fast Food Restaurant	Moroccan Restaurant	Middle Eastern Restaurant
7	Golden Mile, Clairlea, Oakridge	43.711112	-79.284577	4	Mexican Restaurant	Fast Food Restaurant	Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant
8	Birch Cliff, Cliffside West	43.692657	-79.264848	4	Thai Restaurant	Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant	Taiwanese Restaurant
9	Dorset Park, Wexford Heights, Scarborough Town Centre	43.757410	-79.273304	4	Asian Restaurant	Indian Restaurant	Chinese Restaurant	Vietnamese Restaurant	Fast Food Restaurant
10	Wexford, Maryvale	43.750072	-79.295849	4	Indian Chinese Restaurant	Middle Eastern Restaurant	Asian Restaurant	Korean Restaurant	Vietnamese Restaurant
11	Agincourt	43.794200	-79.262029	8	Sri Lankan Restaurant	Malay Restaurant	Filipino Restaurant	Cantonese Restaurant	Chinese Restaurant
12	Clarks Corners, Tam O'Shanter, Sullivan	43.781638	-79.304302	4	Taiwanese Restaurant	Cantonese Restaurant	Caribbean Restaurant	Vietnamese Restaurant	Fast Food Restaurant
13	Milliken, Agincourt North, Steeles East, L'Amoreaux East	43.815252	-79.284577	4	Malay Restaurant	Chinese Restaurant	Caribbean Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant
14	Steeles West, L'Amoreaux West	43.799525	-79.318389	7	Hotpot Restaurant	Chinese Restaurant	Fast Food Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant
15	Hillcrest Village	43.803762	-79.363452	4	Korean Restaurant	Chinese Restaurant	Fast Food Restaurant	Restaurant	Vegetarian / Vegan Restaurant
16	Fairview, Henry Farm, Oriole	43.778517	-79.346556	4	Asian Restaurant	Caribbean Restaurant	American Restaurant	Japanese Restaurant	Fast Food Restaurant
17	Bayview Village	43.786947	-79.385975	4	Japanese Restaurant	Chinese Restaurant	Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant
18	Willowdale, Newtonbrook	43.789053	-79.408493	4	Korean Restaurant	Middle Eastern Restaurant	Ramen Restaurant	Indian Restaurant	Japanese Restaurant

19	Willowdale, Willowdale East	43.770120	-79.408493	4	Indonesian Restaurant	Dumpling Restaurant	Ramen Restaurant	Korean Restaurant	Middle Eastern Restaurant
20	York Mills West	43.752758	-79.400049	4	French Restaurant	Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant	Taiwanese Restaurant
21	Willowdale, Willowdale West	43.782736	-79.442259	4	Eastern European Restaurant	Moroccan Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant	Taiwanese Restaurant
22	Parkwoods	43.753259	-79.329656	4	Caribbean Restaurant	Chinese Restaurant	Fast Food Restaurant	Moroccan Restaurant	Middle Eastern Restaurant
23	Don Mills	43.745906	-79.352188	4	Dim Sum Restaurant	Asian Restaurant	New American Restaurant	Japanese Restaurant	Restaurant
24	Don Mills	43.725900	-79.340923	4	Dim Sum Restaurant	Asian Restaurant	New American Restaurant	Japanese Restaurant	Restaurant
25	Bathurst Manor, Wilson Heights, Downsview North	43.754328	-79.442259	4	Mediterranean Restaurant	Middle Eastern Restaurant	Sushi Restaurant	Restaurant	Ramen Restaurant
26	Northwood Park, York University	43.767980	-79.487262	4	Doner Restaurant	Caribbean Restaurant	Middle Eastern Restaurant	Chinese Restaurant	Fast Food Restaurant
27	Downsview	43.737473	-79.464763	4	Turkish Restaurant	Vietnamese Restaurant	Latin American Restaurant	Falafel Restaurant	Chinese Restaurant
28	Downsview	43.739015	-79.506944	4	Turkish Restaurant	Vietnamese Restaurant	Latin American Restaurant	Falafel Restaurant	Chinese Restaurant
29	Downsview	43.728496	-79.495697	4	Turkish Restaurant	Vietnamese Restaurant	Latin American Restaurant	Falafel Restaurant	Chinese Restaurant
30	Downsview	43.761631	-79.520999	4	Turkish Restaurant	Vietnamese Restaurant	Latin American Restaurant	Falafel Restaurant	Chinese Restaurant
31	Victoria Village	43.725882	-79.315572	4	Portuguese Restaurant	Moroccan Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant	Taiwanese Restaurant
32	Parkview Hill, Woodbine Gardens	43.706397	-79.309937	4	Fast Food Restaurant	Moroccan Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant	Taiwanese Restaurant
33	Woodbine Heights	43.695344	-79.318389	4	Thai Restaurant	Greek Restaurant	Moroccan Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant
34	The Beaches	43.676357	-79.293031	4	Caribbean Restaurant	Mediterranean Restaurant	Ramen Restaurant	Asian Restaurant	French Restaurant
35	Leaside	43.709060	-79.363452	4	Mexican Restaurant	Indian Restaurant	Restaurant	Sushi Restaurant	Japanese Restaurant
36	Thorncliffe Park	43.705369	-79.349372	10	Afghan Restaurant	Turkish Restaurant	Indian Restaurant	Asian Restaurant	Fast Food Restaurant
37	East Toronto, Broadview North (Old East York)	43.685347	-79.338106	4	Ethiopian Restaurant	Dim Sum Restaurant	Greek Restaurant	Turkish Restaurant	American Restaurant
38	The Danforth West, Riverdale	43.679557	-79.352188	0	Cuban Restaurant	Greek Restaurant	Turkish Restaurant	Tapas Restaurant	Falafel Restaurant
39	India Bazaar, The Beaches West	43.668999	-79.315572	13	Indian Chinese Restaurant	Pakistani Restaurant	Indian Restaurant	Fast Food Restaurant	Asian Restaurant
40	Studio District	43.659526	-79.340923	4	Latin American Restaurant	Vietnamese Restaurant	Comfort Food Restaurant	French Restaurant	American Restaurant
41	Davisville North	43.712751	-79.390197	4	Italian Restaurant	Mexican Restaurant	Fast Food Restaurant	Sushi Restaurant	Ramen Restaurant
42	North Toronto West, Lawrence Park	43.715383	-79.405678	4	Mexican Restaurant	Vietnamese Restaurant	Italian Restaurant	Chinese Restaurant	Fast Food Restaurant
43	Davisville	43.704324	-79.388790	3	Syrian Restaurant	Indonesian Restaurant	Italian Restaurant	Sushi Restaurant	Indian Restaurant
44	Moore Park, Summerhill East	43.689574	-79.383160	4	Cantonese Restaurant	German Restaurant	Modern European Restaurant	Thai Restaurant	Italian Restaurant
45	Summerhill West, Rathnelly, South Hill, Forest Hill SE, Deer Park	43.686412	-79.400049	4	Cantonese Restaurant	German Restaurant	Modern European Restaurant	Thai Restaurant	French Restaurant

46	Rosedale	43.679563	-79.377529	4	Filipino Restaurant	Japanese Restaurant	Moroccan Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant
47	St. James Town, Cabbagetown	43.667967	-79.367675	4	Taiwanese Restaurant	Caribbean Restaurant	American Restaurant	Japanese Restaurant	Indian Restaurant
48	Church and Wellesley	43.665860	-79.383160	4	Theme Restaurant	Ethiopian Restaurant	Falafel Restaurant	Ramen Restaurant	Caribbean Restaurant
49	Regent Park, Harbourfront	43.654260	-79.360636	13	Pakistani Restaurant	German Restaurant	Mediterranean Restaurant	Indian Restaurant	Thai Restaurant
50	Garden District, Ryerson	43.657162	-79.378937	4	German Restaurant	Modern European Restaurant	Falafel Restaurant	New American Restaurant	Ramen Restaurant
51	St. James Town	43.651494	-79.375418	4	German Restaurant	Comfort Food Restaurant	New American Restaurant	Seafood Restaurant	American Restaurant
52	Berczy Park	43.644771	-79.373306	4	Comfort Food Restaurant	New American Restaurant	Japanese Restaurant	French Restaurant	Seafood Restaurant
53	Central Bay Street	43.657952	-79.387383	4	Modern European Restaurant	Falafel Restaurant	Ramen Restaurant	Seafood Restaurant	Mexican Restaurant
54	Richmond, Adelaide, King	43.650571	-79.384568	4	Brazilian Restaurant	New American Restaurant	Mediterranean Restaurant	American Restaurant	Asian Restaurant
55	Harbourfront East, Union Station, Toronto Islands	43.640816	-79.381752	4	Mediterranean Restaurant	Japanese Restaurant	Vegetarian / Vegan Restaurant	French Restaurant	Seafood Restaurant
56	Toronto Dominion Centre, Design Exchange	43.647177	-79.381576	4	Brazilian Restaurant	New American Restaurant	Seafood Restaurant	Mediterranean Restaurant	American Restaurant
57	Commerce Court, Victoria Hotel	43.648198	-79.379817	4	Seafood Restaurant	New American Restaurant	Mediterranean Restaurant	American Restaurant	Japanese Restaurant
58	Bedford Park, Lawrence Manor East	43.733283	-79.419750	4	Comfort Food Restaurant	Greek Restaurant	Italian Restaurant	American Restaurant	Indian Restaurant
59	Roselawn	43.711695	-79.416936	4	Sushi Restaurant	Italian Restaurant	Japanese Restaurant	Moroccan Restaurant	Korean Restaurant
60	Forest Hill North & West, Forest Hill Road Park	43.696948	-79.411307	4	Persian Restaurant	Sushi Restaurant	Middle Eastern Restaurant	Japanese Restaurant	Vegetarian / Vegan Restaurant
61	The Annex, North Midtown, Yorkville	43.672710	-79.405678	4	Jewish Restaurant	Modern European Restaurant	Eastern European Restaurant	Latin American Restaurant	Vegetarian / Vegan Restaurant
62	University of Toronto, Harbord	43.662696	-79.400049	14	Belgian Restaurant	Doner Restaurant	Persian Restaurant	Eastern European Restaurant	Comfort Food Restaurant
63	Kensington Market, Chinatown, Grange Park	43.653206	-79.400049	14	Belgian Restaurant	Filipino Restaurant	Dumpling Restaurant	Doner Restaurant	Persian Restaurant
64	Stn A PO Boxes	43.646435	-79.374846	4	Comfort Food Restaurant	Seafood Restaurant	New American Restaurant	American Restaurant	Japanese Restaurant
65	First Canadian Place, Underground city	43.648429	-79.382280	4	Brazilian Restaurant	New American Restaurant	Seafood Restaurant	Mediterranean Restaurant	American Restaurant
66	Lawrence Manor, Lawrence Heights	43.718518	-79.464763	4	Vietnamese Restaurant	Fast Food Restaurant	Korean Restaurant	Seafood Restaurant	Greek Restaurant
67	Glencairn	43.709577	-79.445073	4	Latin American Restaurant	Mediterranean Restaurant	Fast Food Restaurant	Asian Restaurant	Italian Restaurant
68	Humewood-Cedarvale	43.693781	-79.428191	12	Israeli Restaurant	Korean Restaurant	Mexican Restaurant	Middle Eastern Restaurant	Italian Restaurant
69	Caledonia-Fairbanks	43.689026	-79.453512	4	Portuguese Restaurant	Falafel Restaurant	Mexican Restaurant	Fast Food Restaurant	Japanese Restaurant
70	Christie	43.669542	-79.422564	2	South American Restaurant	Jewish Restaurant	Korean Restaurant	Ethiopian Restaurant	Latin American Restaurant
71	Dufferin, Dovercourt Village	43.669005	-79.442259	4	Portuguese Restaurant	Brazilian Restaurant	Mediterranean Restaurant	Sushi Restaurant	Vietnamese Restaurant
72	Little Portugal, Trinity	43.647927	-79.419750	0	Dumpling Restaurant	Malay Restaurant	Tapas Restaurant	Cuban Restaurant	Asian Restaurant
73	Brockton, Parkdale Village, Exhibition Place	43.636847	-79.428191	1	North Indian Restaurant	Tibetan Restaurant	Hawaiian Restaurant	Tapas Restaurant	Ethiopian Restaurant
74	North Park, Maple Leaf Park, Upwood Park	43.713756	-79.490074	4	Dim Sum Restaurant	Mediterranean Restaurant	Chinese Restaurant	Moroccan Restaurant	Korean Restaurant

75	Del Ray, Mount Dennis, Keelsdale and Silverthorn	43.691116	-79.476013	4	Fast Food Restaurant	Italian Restaurant	Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant
76	Runnymede, The Junction North	43.673185	-79.487262	4	Asian Restaurant	Vietnamese Restaurant	Indian Restaurant	Thai Restaurant	Fast Food Restaurant
77	High Park, The Junction South	43.661608	-79.464763	6	Cajun / Creole Restaurant	Thai Restaurant	Mediterranean Restaurant	Mexican Restaurant	Sushi Restaurant
78	Parkdale, Roncesvalles	43.648960	-79.456325	4	Eastern European Restaurant	Cuban Restaurant	Falafel Restaurant	Mediterranean Restaurant	Thai Restaurant
79	Runnymede, Swansea	43.651571	-79.484450	4	Falafel Restaurant	Latin American Restaurant	Sushi Restaurant	French Restaurant	Greek Restaurant
80	Queen's Park, Ontario Provincial Government	43.662301	-79.389494	4	Theme Restaurant	Modern European Restaurant	Ethiopian Restaurant	Ramen Restaurant	Falafel Restaurant
81	Canada Post Gateway Processing Centre	43.636966	-79.615819	4	Portuguese Restaurant	Falafel Restaurant	Asian Restaurant	Middle Eastern Restaurant	Mexican Restaurant
82	Business reply mail Processing Centre, South Central Letter Processing Plant Toronto	43.662744	-79.321558	4	French Restaurant	Fast Food Restaurant	Sushi Restaurant	American Restaurant	Thai Restaurant
83	New Toronto, Mimico South, Humber Bay Shores	43.605647	-79.501321	4	Mexican Restaurant	American Restaurant	Indian Restaurant	Fast Food Restaurant	Italian Restaurant
84	Alderwood, Long Branch	43.602414	-79.543484	11	Moroccan Restaurant	Ramen Restaurant	Vegetarian / Vegan Restaurant	Taiwanese Restaurant	Filipino Restaurant
85	The Kingsway, Montgomery Road, Old Mill North	43.653654	-79.506944	4	Tapas Restaurant	French Restaurant	Seafood Restaurant	Greek Restaurant	Sushi Restaurant
86	Old Mill South, King's Mill Park, Sunnylea, Humber Bay, Mimico NE, The Queensway East, Royal York South East, Kingsway Park South East	43.636258	-79.498509	4	Eastern European Restaurant	Italian Restaurant	Moroccan Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant
87	Mimico NW, The Queensway West, South of Bloor, Kingsway Park South West, Royal York South West	43.628841	-79.520999	4	Comfort Food Restaurant	Mediterranean Restaurant	Asian Restaurant	Restaurant	Mexican Restaurant
88	West Deane Park, Princess Gardens, Martin Grove, Islington, Cloverdale	43.650943	-79.554724	4	Mexican Restaurant	Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant	Taiwanese Restaurant
89	Humber Summit	43.756303	-79.565963	4	Italian Restaurant	Moroccan Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant	Taiwanese Restaurant
90	Weston	43.706876	-79.518188	4	Middle Eastern Restaurant	Moroccan Restaurant	Ramen Restaurant	Vegetarian / Vegan Restaurant	Taiwanese Restaurant
91	Westmount	43.696319	-79.532242	4	Middle Eastern Restaurant	Chinese Restaurant	Moroccan Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant
92	Kingsview Village, St. Phillips, Martin Grove Gardens, Richview Gardens	43.688905	-79.554724	4	American Restaurant	Chinese Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant	Taiwanese Restaurant
93	South Steeles, Silverstone, Humbergate, Jamestown, Mount Olive, Beaumond Heights, Thistletown, Albion Gardens	43.739416	-79.588437	4	Caribbean Restaurant	Fast Food Restaurant	Moroccan Restaurant	Korean Restaurant	Vegetarian / Vegan Restaurant