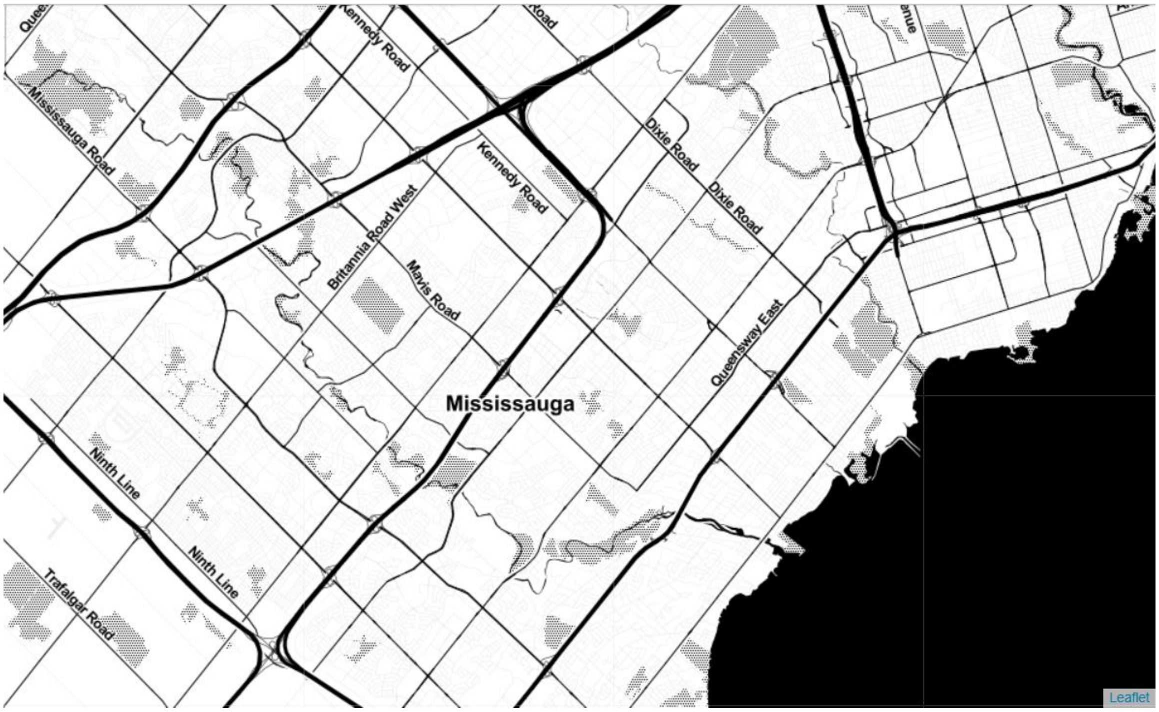


The Battle of Neighborhoods

PROJECT REPORT

Olga Tolmacheva | Applied Data Science Capstone | 2019

Introduction/Business Problem

The objective of this project is to apply Data Science techniques and select a neighborhood in the city of Mississauga, Canada, which is well suited to open a new Pizza Restaurant.

Target audience/potential stakeholders for this project are:

- Entrepreneurs interested in launching new Pizza business;
- Established Pizza businesses looking to expand their brand in Mississauga.

Data

For the execution of this project, the following data sources will be used:

List of Mississauga neighborhoods. This list will be retrieved from Wikipedia page (https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_L), using web scraping techniques.


Not logged in Talk Contributions Create account Log in

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Article Talk Read Edit View history Search Wikipedia

List of postal codes of Canada: L

From Wikipedia, the free encyclopedia

This is a list of postal codes in Canada where the first letter is L. Postal codes beginning with L are located within the Canadian province of Ontario. Only the first three characters are listed, corresponding to the Forward Sortation Area.

Canada Post provides a free postal code look-up tool on its website,[1] via its smartphone applications for iPhone and BlackBerry,[2] and sells hard-copy directories and CD-ROMs. Many vendors also sell validation tools, which allow customers to properly match addresses and postal codes. Hard-copy directories can also be consulted in all post offices, and some libraries.

Central Ontario - 163 FSAs [edit]

Urban [edit]

L1A Port Hope	**L2A** Fort Erie	L3A *Not assigned*	**L4A** Stouffville	**L5A** Mississauga (Mississauga Valley / East Cooksville)	**L6A** Maple	**L7A** Brampton West	L8A *Not assigned*	**L9A** Hamilton (Crerar / Bruleville / Hill Park / Inch Park / Centremount / Balfour / Greeningdon / Jerome)
L1B Bowmanville East	L2B *Not assigned*	**L3B** Welland East	**L4B** Richmond Hill Southeast	**L5B** Mississauga (West Cooksville /	**L6B** Markham East Cornell,	**L7B** King City	**L8B** Waterdown	**L9B** Hamilton (Barnstown / West Chappel /

Geographical coordinates of the neighborhoods. This data will be retrieved using *OpenStreetMap Nominatim* via *geopy* library.

```
# install geopy and import Nominatim for working with geo coordinates
!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim
```

...

```
geo=geolocator.geocode("Mississauga Valley, Mississauga, ON")
print(geo.latitude)
print(geo.longitude)
```

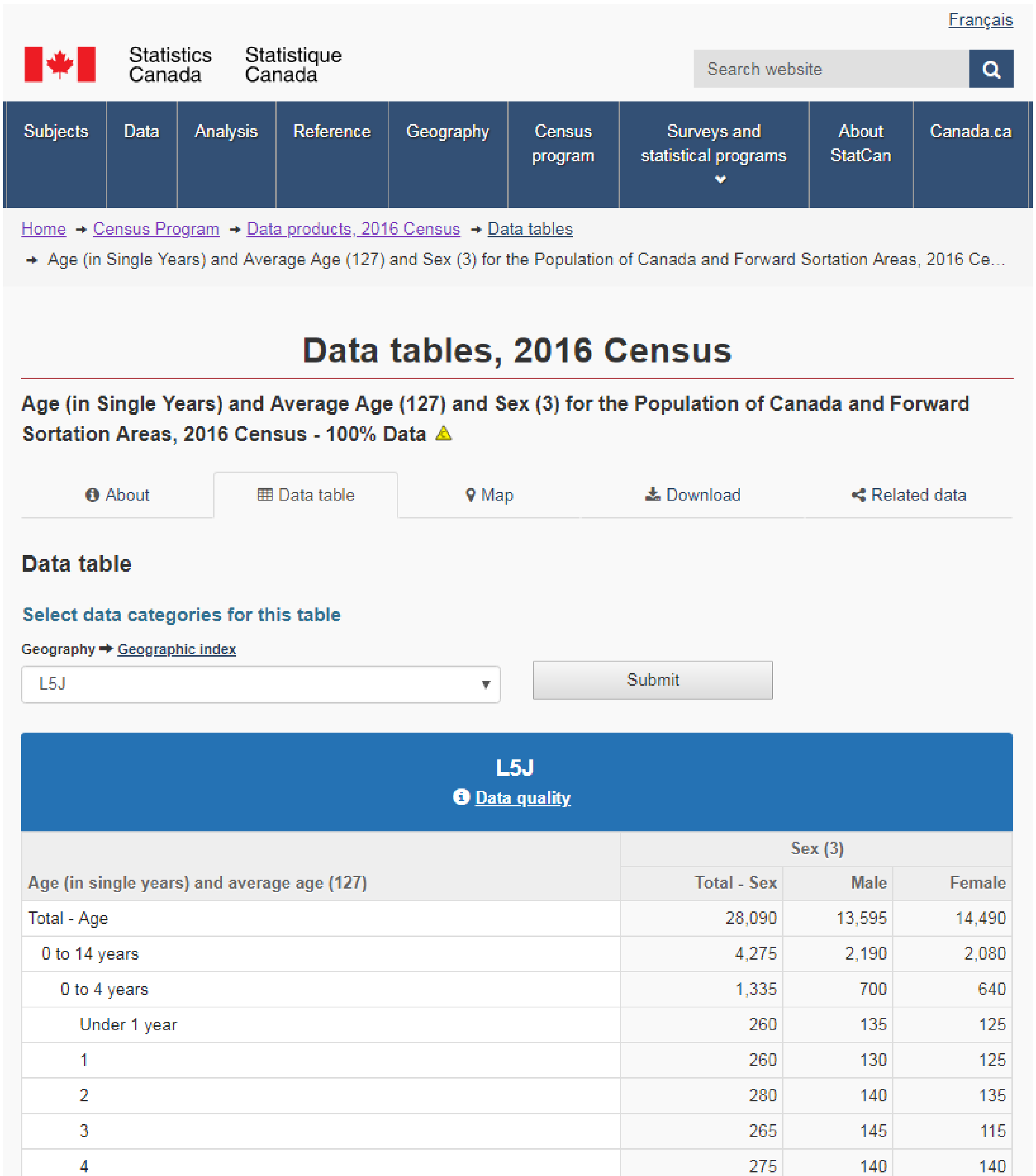
```
43.5943679
-79.6237667
```

Foursquare location data. This data will be retrieved using Foursquare API. We will be using *Explore* function to retrieve nearby venues, given geographic coordinates of the neighborhoods.

```
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
            CLIENT_ID, CLIENT_SECRET, VERSION, geo.latitude, geo.longitude, 500, 100)
results = requests.get(url).json()
results
```


```
{'meta': {'code': 200, 'requestId': '5c31733d6a60713c2bd9d26b'},
 'response': {'headerLocation': 'Cooksville',
  'headerFullLocation': 'Cooksville, Mississauga',
  'headerLocationGranularity': 'neighborhood',
  'totalResults': 11,
  'suggestedBounds': {'ne': {'lat': 43.598867904500004,
    'lng': -79.6175648820002},
   'sw': {'lat': 43.5898678955, 'lng': -79.62996851799981}},
  'groups': [{'type': 'Recommended Places',
    'name': 'recommended',
    'items': [{'reasons': {'count': 0,
       'items': [{'summary': 'This spot is popular',
         'type': 'general',
         'reasonName': 'globalInteractionReason'}]},
      'venue': {'id': '4b0a06b6f964a5202f2123e3',
       'name': 'Mississauga Valley Park',
       'location': {'address': '1275 MIssissauga Valley Blvd',
        'lat': 43.596026195380325,
        'lng': -79.62405681610107,
        'labeledLatLngs': [{'label': 'display',
          'lat': 43.596026195380325,
          'lng': -79.62405681610107}],
        'distance': 186,
        'postalCode': 'L5A 3S8',
        'cc': 'CA',
        'city': 'Mississauga',
        'state': 'ON',
        'country': 'Canada',
        'formattedAddress': ['1275 MIssissauga Valley Blvd',
```

Population Age by Postal Code. This information will be retrieved from Statistics Canada website (https://www.statcan.gc.ca/eng/start) from the results of the 2016 Census, using web scraping techniques and used to provide insight into demographic makeup of the candidate neighborhoods.



Français

Statistics Canada Statistique Canada

Search website

Subjects | Data | Analysis | Reference | Geography | Census program | Surveys and statistical programs | About StatCan | Canada.ca

Home → Census Program → Data products, 2016 Census → Data tables

→ Age (in Single Years) and Average Age (127) and Sex (3) for the Population of Canada and Forward Sortation Areas, 2016 Ce...

Data tables, 2016 Census

Age (in Single Years) and Average Age (127) and Sex (3) for the Population of Canada and Forward Sortation Areas, 2016 Census - 100% Data

About | Data table | Map | Download | Related data

Data table

Select data categories for this table

Geography → Geographic index

L5J

Submit

L5J

Data quality

	Sex (3)		
Age (in single years) and average age (127)	Total - Sex	Male	Female
Total - Age	28,090	13,595	14,490
0 to 14 years	4,275	2,190	2,080
0 to 4 years	1,335	700	640
Under 1 year	260	135	125
1	260	130	125
2	280	140	135
3	265	145	115
4	275	140	140

Methodology

Data Capture

First, we will obtain our data.

The **list of Mississauga neighborhoods** will be retrieved from the Wikipedia page.

We will be using *Beautiful Soup* python library and *lxml* parser for this task:

```
# get the neighborhoods data wiki page and parse content with BeautifulSoup and lxml parser
url = 'https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_L'
page = requests.get(url)
table = BeautifulSoup(page.content, 'lxml').find('table')
print("Wiki page data captured.")
```

```
Wiki page data captured.
```

Now we can extract neighborhood names and retrieve their **geographical coordinates** using *geopy* with *Nominatim*.

Results will be stored in *pandas* Dataframe.

We can also go ahead and visualize our neighborhoods on a map, using *Folium* library:

```
We have retrieved coordinates of 35 neighborhoods
```

	index	PostalCode	Neighborhood	Latitude	Longitude
0	0	L5A	Mississauga Valley	43.594368	-79.623767
1	1	L5A	East Cooksville	43.580244	-79.616376
2	3	L5B	Fairview	43.581089	-79.635256
3	4	L5B	City Centre	43.588499	-79.644108
4	5	L5B	East Creditview	43.606185	-79.723675

...

	index	PostalCode	Neighborhood	Latitude	Longitude
30	37	L5W	Meadowvale Village	43.627081	-79.727791
31	39	L4X	East Applewood	43.648027	-79.568764
32	40	L4X	East Dixie	43.603386	-79.590415
33	46	L4Z	East Hurontario	43.630286	-79.685086
34	48	L4Z	Sandalwood	43.614735	-79.661563

Next, we will use Foursquare API to retrieve information about venues in our neighborhoods. Results will be stored in *pandas* Dataframe:

```
We have retrieved 407 venues in 33 neighborhoods from Foursquare API
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Mississauga Valley	43.594368	-79.623767	Mississauga Valley Park	43.596026	-79.624057	Park
1	Mississauga Valley	43.594368	-79.623767	Mississauga Valley Community Centre	43.596918	-79.624388	Gym / Fitness Center
2	Mississauga Valley	43.594368	-79.623767	Shoppers Drug Mart	43.592338	-79.626280	Pharmacy
3	Mississauga Valley	43.594368	-79.623767	Subway	43.591915	-79.627075	Sandwich Place
4	Mississauga Valley	43.594368	-79.623767	TD Canada Trust	43.592410	-79.626452	Bank

Data Analysis

Immediately we observe that only 33 neighborhoods out of 35 were returned from Foursquare API.

This is not a concern, because for our goal we wouldn't be interested in locations that haven't had any places of interest on Foursquare.

Consequently, we can now proceed with our analysis of the 33 neighborhoods.

We will start with exploring what unique venue categories were present in our data capture:

```
city_venues['Venue Category'].unique()
```

Here are the results:

```
array(['Park', 'Gym / Fitness Center', 'Pharmacy', 'Sandwich Place',
       'Bank', 'Tennis Court', 'Pizza Place', 'Skating Rink', 'Plaza',
       'Convenience Store', 'Trail', 'Burrito Place', 'Korean Restaurant',
       'Indian Restaurant', 'Caribbean Restaurant',
       'Middle Eastern Restaurant', 'Café', 'Vietnamese Restaurant',
       'Portuguese Restaurant', 'Grocery Store', 'Fried Chicken Joint',
       'BBQ Joint', 'Mexican Restaurant', 'Fast Food Restaurant',
       'Chinese Restaurant', 'Mediterranean Restaurant', 'Supermarket',
       'Pakistani Restaurant', 'Paper / Office Supplies Store',
       'Coffee Shop', 'Bakery', 'Sushi Restaurant', 'Shopping Mall',
       'Bus Station', 'Train', 'Field', 'Performing Arts Venue',
       'Italian Restaurant', 'Burger Joint', 'Yoga Studio',
       'Electronics Store', 'College Gym', 'Sporting Goods Shop',
       'Cosmetics Shop', 'Department Store', 'Clothing Store',
```

```
'Discount Store', 'Wings Joint', 'Deli / Bodega', 'Spa', 'Gym',
'Event Service', 'Mountain', 'Construction & Landscaping', 'Farm',
'Hotel', 'Bar', 'Restaurant', 'Pool Hall', 'Irish Pub',
'Ice Cream Shop', 'Tapas Restaurant', 'Poutine Place',
'Harbor / Marina', 'Gastropub', 'Cocktail Bar',
'Seafood Restaurant', 'Gas Station', 'Breakfast Spot',
'Waterfront', 'Gift Shop', 'Motel', 'Health & Beauty Service',
'Gun Range', 'Liquor Store', 'Japanese Restaurant',
'Video Game Store', 'Salon / Barbershop', 'Platform',
'Train Station', 'Falafel Restaurant', 'Bookstore',
'Asian Restaurant', 'Tea Room', 'Beer Store', 'Optical Shop',
'Pet Store', 'Record Shop', 'Shoe Store', "Women's Store",
'Rental Car Location', 'Baseball Field', 'Airport',
'Airport Terminal', 'Vegetarian / Vegan Restaurant', 'Diner',
'Greek Restaurant', 'American Restaurant', 'Building',
'Big Box Store', 'Transportation Service', 'Movie Theater',
'New American Restaurant', 'Thai Restaurant', 'Dessert Shop',
'Steakhouse', 'Smoothie Shop', 'Salad Place', 'Athletics & Sports',
'Afghan Restaurant', 'Shop & Service', 'Shopping Plaza',
'Mobile Phone Shop', 'Pool', 'Donut Shop', 'Beer Bar',
'Kids Store', 'Furniture / Home Store', 'Bubble Tea Shop',
'Smoke Shop', 'Candy Store', 'Playground', 'Sports Club',
'History Museum', 'Laser Tag', 'Thrift / Vintage Store',
'Light Rail Station', 'Road'], dtype=object)
```

Since our objective is to find a suitable location for a Pizza Place, we are mostly interested in foodservice venues, such as fast food joints or restaurants.

With that goal in mind, we will classify Foursquare venue categories in three sets: Fast Food, Restaurant and Other Venue:

- If venue category contains words like ‘Pizza’, ‘Burger’, ‘Sandwich’, then it will be assigned Restaurant Category **Fast Food**;
- If venue category contains word ‘Restaurant’, it will be assigned to **Restaurant** category;
- The rest of the venue categories can be classified as **Other Venue**.

We will update our venues dataframe accordingly:

```
city_venues['Restaurant Category'] = 'Other Venue'
city_venues.loc[city_venues['Venue Category'].str.contains('Pizza') |
                city_venues['Venue Category'].str.contains('Burger') |
                city_venues['Venue Category'].str.contains('Sandwich'), 'Restaurant Category'] = 'Fast Food'
city_venues.loc[city_venues['Venue Category'].str.contains('Restaurant'), 'Restaurant Category'] = 'Restaurant'
city_venues.head()
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category	Restaurant Category
0	Mississauga Valley	43.594368	-79.623767	Mississauga Valley Park	43.596026	-79.624057	Park	Other Venue
1	Mississauga Valley	43.594368	-79.623767	Mississauga Valley Community Centre	43.596918	-79.624388	Gym / Fitness Center	Other Venue
2	Mississauga Valley	43.594368	-79.623767	Shoppers Drug Mart	43.592338	-79.626280	Pharmacy	Other Venue
3	Mississauga Valley	43.594368	-79.623767	Subway	43.591915	-79.627075	Sandwich Place	Fast Food
4	Mississauga Valley	43.594368	-79.623767	TD Canada Trust	43.592410	-79.626452	Bank	Other Venue

Data Segmentation

At this point we can start on data segmentation using k-means clustering techniques.

k-means clustering methodology will allow us to segment all candidate neighborhoods into 'alike' clusters with respect to distribution of foodservice locations vs other venues.

First, we need to prepare a dataset suitable for k-means algorithm.

We will calculate distribution levels by neighborhood using **Restaurant Category** field and dummy variables technique. Here is our dataset ready for clustering:

```
The shape of our dataset for clustering is: (33, 4)
```

	Neighborhood	Fast Food	Other Venue	Restaurant
0	Central Erin Mills	0.041667	0.791667	0.166667
1	Central Lakeview	0.000000	1.000000	0.000000
2	Churchill Meadows	0.181818	0.727273	0.090909
3	City Centre	0.040000	0.920000	0.040000
4	Clarkson	0.107143	0.714286	0.178571

We will now execute k-means clustering, setting a number of clusters to 5:

```
# cluster
kclusters = 5
df_clustering = df_grouped.drop('Neighborhood', 1)
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(df_clustering)
kmeans.labels_
```

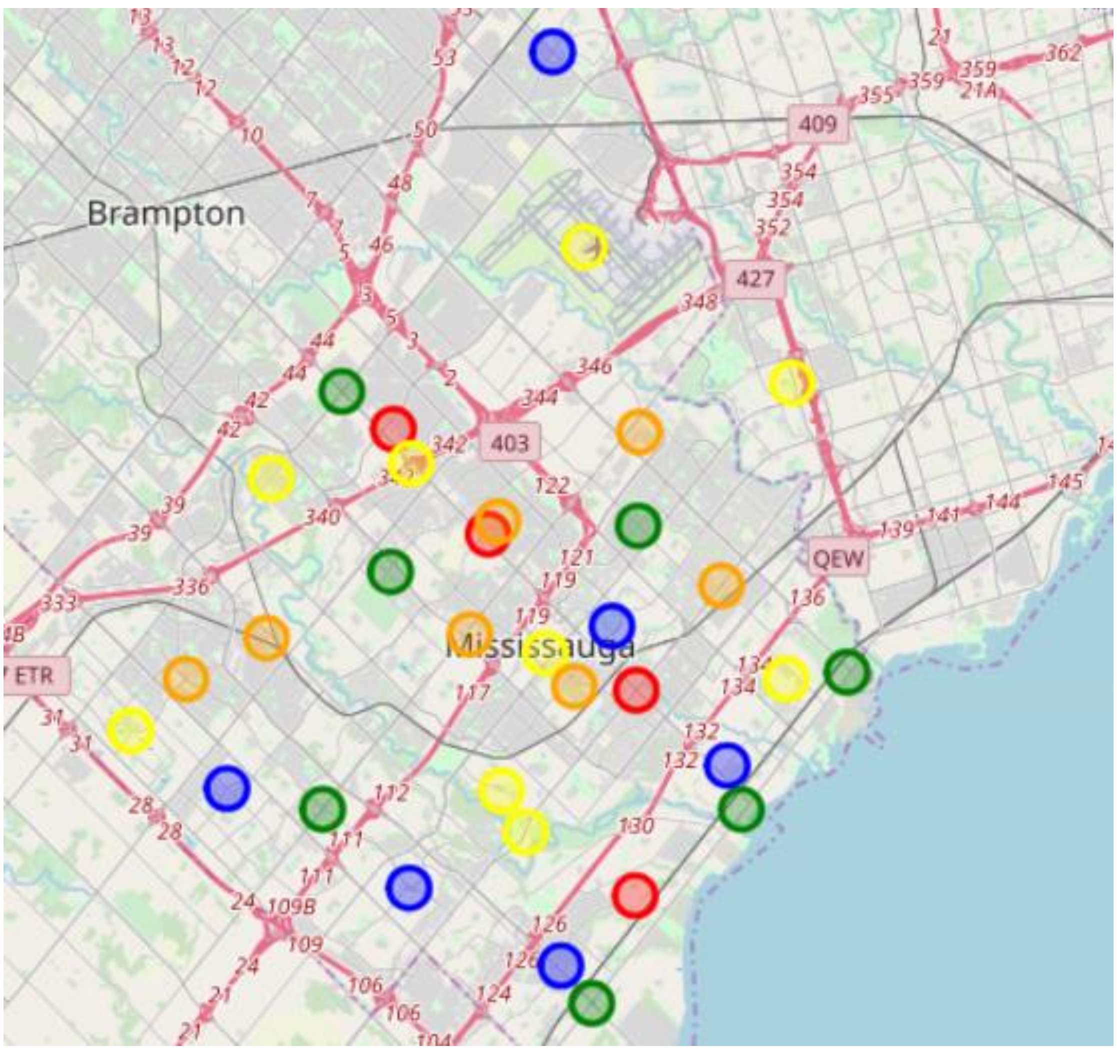
```
array([0, 2, 4, 2, 0, 1, 2, 1, 2, 3, 2, 0, 0, 0, 4, 2, 3, 2, 1, 4, 0, 3,
       3, 2, 4, 4, 3, 1, 3, 4, 0, 3, 2], dtype=int32)
```

Excellent! k-means model produced 5 clusters of neighborhoods with similar characteristics.

We can add the resulting cluster labels with respective distribution levels back to our dataframe of neighborhoods:

	Neighborhood	Fast Food	Other Venue	Restaurant	Label	PostalCode	Latitude	Longitude
0	Central Erin Mills	0.041667	0.791667	0.166667	0	L5M	43.553932	-79.712080
1	Central Lakeview	0.000000	1.000000	0.000000	2	L5E	43.582657	-79.570649
2	Churchill Meadows	0.181818	0.727273	0.090909	4	L5M	43.558520	-79.741275
3	City Centre	0.040000	0.920000	0.040000	2	L5B	43.588499	-79.644108
4	Clarkson	0.107143	0.714286	0.178571	0	L5J	43.511044	-79.629934

Let's go ahead and visualize our clusters on a map:



And now let’s examine the clusters.

	Label	Fast Food	Other Venue	Restaurant
0	4	0.188582	0.796266	0.015152
1	3	0.085826	0.591588	0.322586
2	1	0.075423	0.442082	0.482495
3	0	0.058940	0.739046	0.202014
4	2	0.004444	0.991111	0.004444

We will group clustering dataframe by cluster label and calculate average distribution levels for our restaurant categories.

From the resulting dataframe we can take a look at cluster characteristics at-a-glance.

We observe that one of the clusters (Cluster 2) presents no interest to us due to the fact that there is effectively no interest in foodservice locations in neighborhoods of this cluster: 99% of all Foursquare venues were categorized as Other Venue. We can eliminate this cluster from further analysis.

We also observe that Cluster 1 has clear preference for foodservice locations in Restaurant category (48% of all points of interest), while very little interest in Fast Food (8%). We will eliminate this cluster as well.

Of the remaining clusters, Cluster 4 clearly stands out as a candidate for further analysis due to significantly higher interest in Fast Food venues compared to other clusters (19% vs 9% and 6%), where the interest is higher in foodservice locations of Restaurant category (32% and 20%).

Having identified our target cluster, we can now proceed with further analysis.

Let's take a closer look at Cluster 4:

```
#filter by Label and sort
top_label = df_clusters['Label'][0]
df_focus = df_labeled[(df_labeled['Label']==top_label)].sort_values(by='Fast Food', ascending = False).reset_index(drop = True)
print("Top cluster by Fast Food is Cluster " + str(top_label) + ". It contains " + str(df_focus.shape[0]) + " neighborhoods  \n")
df_focus.head()
```

```
Top cluster by Fast Food is Cluster 4. It contains 6 neighborhoods
```

	Neighborhood	Fast Food	Other Venue	Restaurant	Label	PostalCode	Latitude	Longitude
0	Mineola	0.250000	0.750000	0.000000	4	L5G	43.563617	-79.588591
1	Southdown	0.250000	0.750000	0.000000	4	L5J	43.519366	-79.639351
2	Churchill Meadows	0.181818	0.727273	0.090909	4	L5M	43.558520	-79.741275
3	Mississauga Valley	0.181818	0.818182	0.000000	4	L5A	43.594368	-79.623767
4	Malton	0.142857	0.857143	0.000000	4	L4T	43.721070	-79.641868

The cluster contains 6 neighborhoods.

Evidently, the outcome of our analysis is too broad to present to the stakeholders.

Let's see if we can refine our results by retrieving and examining additional information about these neighborhoods.

Additional Analysis

We will refer to *Global Consumer Trends* Market Analysis Report published in 2012 by Agriculture and Agri-Food Canada.

This report suggests that population demographic characteristics, in particular Age Group, can be useful in predicting consumer trends in Agriculture and Food space.

It may be possible then to narrow down the list of our neighborhoods by exploring their demographic characteristics.

Let's start by capturing relevant data.

We will be using *Beautiful Soup* python library and *lxml* parser to access Statistics Canada web site and retrieve information about population age from the Census 2016 data tables.

This data is available by Forward Sortation Area (the first three characters of the Postal Code) and represents population counts in that area for each Age year.

As we process the web page content, we will record age of 0-12 months as **1** (year) and age over 100 years as **100** (years old) for ease of further analysis.

Results will be stored in *pandas* dataframe.

	PostalCode	Age	Population
0	L4T	1	450
1	L4T	1	450
2	L4T	2	455
3	L4T	3	500
4	L4T	4	465

...

	PostalCode	Age	Population
1712	L5W	96	0
1713	L5W	97	5
1714	L5W	98	0
1715	L5W	99	0
1716	L5W	100	5

Let's use binning technique on this data to create Age Groups.

	PostalCode	Age	Population	AgeGroup
0	L4T	1	450	Babies and Toddlers
1	L4T	1	450	Babies and Toddlers
2	L4T	2	455	Babies and Toddlers
3	L4T	3	500	Babies and Toddlers
4	L4T	4	465	Babies and Toddlers

Babies and Toddlers: 0-4 years
Children: 5-12 years
Teens: 13-18 years
Youth: 18-24 years
Young Adults: 25-34 years
Adults: 35-64 years
Younger Seniors: 64-80 years
Older Seniors: 80+ years

This approach will make age data more meaningful for further analysis.

Now we can aggregate the population counts and then calculate percentage of population by Postal Code and Age Group:

	PostalCode	AgeGroup	AgeGroupTotal	AreaTotal	AgeGroupPct
0	L4T	Babies and Toddlers	2320	38445	0.060346
1	L4T	Children	4230	38445	0.110027
2	L4T	Teens	3010	38445	0.078294
3	L4T	Youth	3515	38445	0.091429
4	L4T	Young Adults	5605	38445	0.145793

From the *Global Consumer Trends* report we know that consumption of ready-made meals, and specifically, spending on take-out from foodservice outlets is higher for the consumers in age groups from Teens to Young Adults.

With that information in mind, we will define our target consumer group as Teens, Youth and Young Adults and proceed to calculate total share of these age groups within each Postal Code area.

	PostalCode	AgeGroup	AgeGroupTotal	AreaTotal	AgeGroupPct
0	L4T	Babies and Toddlers	2320	38445	0.060346
1	L4T	Children	4230	38445	0.110027
2	L4T	Teens	3010	38445	0.078294
3	L4T	Youth	3515	38445	0.091429
4	L4T	Young Adults	5605	38445	0.145793

	TargetAgePopulation	TargetAgeShare
PostalCode		
L4T	12130	0.315516
L4W	4985	0.240125
L4X	5475	0.279265
L4Z	11960	0.314034
L5A	13310	0.278190

We will then combine this information with the data on our target cluster and normalize data in fields of interest for further analysis:

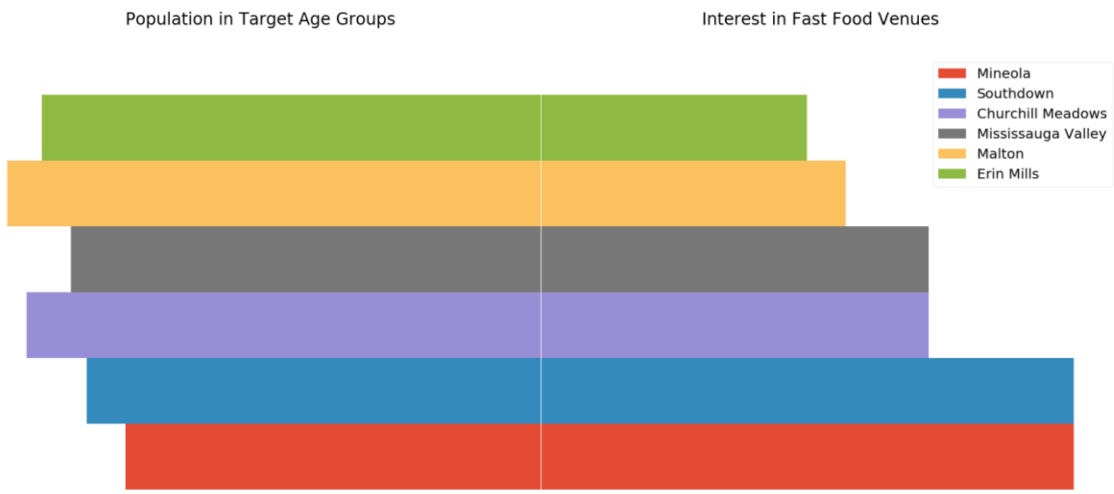
```
# merge new data into target Cluster dataframe
df_merged = pd.merge(df_focus, df_target_pop, left_on = 'PostalCode', right_on = 'PostalCode')

# normalize data
df_merged['FastFoodPref'] = df_merged['Fast Food']/df_merged['Fast Food'].max()
df_merged['TargetAge'] = df_merged['TargetAgeShare'] / df_merged['TargetAgeShare'].max()

# compose results
df_results = df_merged[['Neighborhood', 'FastFoodPref', 'TargetAge']].copy()
df_results.head()
```

	Neighborhood	FastFoodPref	TargetAge
0	Mineola	1.000000	0.779197
1	Southdown	1.000000	0.851266
2	Churchill Meadows	0.727273	0.963882
3	Mississauga Valley	0.727273	0.881699
4	Malton	0.571429	1.000000

Let’s plot normalized data using *matplotlib* and visually examine how individual neighborhoods within our target cluster compare to each other.



It appears from the bar plot that the "ideal" neighborhood is **Southdown**, due to higher values of both, the share of population in our target Age Group, and the preference for Fast Food locations, compared to other venues.

Summing up values for both characteristics in our dataframe and selecting the neighborhood with top ‘Score’ confirms this observation:

```
df_results['Score'] = df_results['FastFoodPref'] + df_results['TargetAge']
df_top = df_results.sort_values(by='Score', ascending = False)
df_top.head(1)
```

	Neighborhood	FastFoodPref	TargetAge	Score
1	Southdown	1.0	0.851266	1.851266

Results

As the result of our analysis we can recommend Southdown neighborhood as a good candidate for a new Pizza Place in Mississauga.

Observations and Recommendations

We have observed that location data providers, such as *Foursquare*, combined with the machine learning techniques, such as data segmentation using *k-means* model, can be extremely helpful when solving problems related to geographical locations.

At the same time, we have observed that relying strictly on Foursquare data may not deliver results precise enough for a given problem. It is our recommendation therefore, that additional sources of information should be considered when approaching problems related to geographic locations.

In our example, Age Demographics information provided additional insights fthat helped with the final decision.

Conclusion

In conclusion, we were able to successfully perform Data Analysis by applying Data Science techniques to select a neighborhood in the city of Mississauga, which is well suited to open a new Pizza Restaurant.

References

1. Wikipedia contributors. List of postal codes of Canada: L. Wikipedia, The Free Encyclopedia. November 28, 2018, 17:26 UTC. Available at: https://en.wikipedia.org/w/index.php?title=List_of_postal_codes_of_Canada:_L&oldid=871056773. Accessed January 6, 2019.

2. Statistics Canada. Census Program - Data products, 2016 Census - Data tables. Age (in Single Years) and Average Age (127) and Sex (3) for the Population of Canada and Forward Sortation Areas, 2016 Census - 100% Data. Available at: https://www12.statcan.gc.ca/census-recensement/2016/dp-pd/dt-td/Rp-eng.cfm?TABID=6&LANG=E&APATH=3&DETAIL=0&DIM=0&FL=A&FREE=0&GC=A0A&GID=1256482&GK=9&GRP=1&PID=109790&PRID=10&PTYPE=109445&S=0&SHOWALL=0&SUB=0&Temporal=2016&THEME=115&VID=0&VNAMEE=&VNAMEF=&D1=0&D2=0&D3=0&D4=0&D5=0&D6=0. Accessed January 6, 2019.

3. Agriculture and Agri-Food Canada. Global Consumer Trends – Market Analysis Report 2012. Available at: http://publications.gc.ca/collections/collection_2012/agr/A74-2-2012-11-eng.pdf. Accessed Jan 13, 2019