Capstone Project – Final Submission

Introduction

a. Background

The final assessment set by the Coursera Applied Data Science Capstone centres around leveraging Foursquare location data to explore and compare neighbourhoods or cities of my choice; and to come up with and solve a problem using Foursquare location data.

b. Problem

The task can be split into two sections: neighbourhood/city comparison, and problem solving hereafter named Question 1 and Question 2 respectively.

Question 1 objective: To compare the neighbourhoods of Downtown Toronto and Ottawa and determine how similar or dissimilar they are.

Using the Foursquare API I will explore the most common venue categories in Downtown Toronto and Ottawa, then use this feature to group the neighbourhoods into clusters – using K means. After which I will use the Folium library to visualise the neighbourhoods in both Toronto and Ottawa along with their emerging venue clusters. This information will benefit the Government of Canada as they are attempting to establish the diversity of venue types in these locations.

Question 2 objective: A restaurant owner is looking to open a new Italian restaurant in Toronto, the objective is to recommend the best area in which a new restaurant could be located.

Using the Foursquare API I will explore the Italian restaurants in each neighbourhood in Toronto. After which I will use the Folium library to visualise the restaurants to inform the owner of the current distribution. The spatial distribution is highly important from a competition point of view as an area highly saturated in italian cuisine will prove detrimental to their business. Therefore, the owner will be looking for an area that has none/few Italian restaurants at present.

c. Interest

As mentioned in the problem section of this report the interest can also be split into two sections.

Question 1 interest: The target audience of this analysis are the analysts within the Government of Canada. The stakeholders are the wider Government of Canada.

Question 2 interest: The target audience is the restaurant owner. The stakeholders are the bank – who are lending the owner the money to build his new restaurant.

Data acquisition and cleaning

a. Data sources

Question 1 data:

Csv of Toronto location data containing postal codes, boroughs and neighbourhoods – from a previous assessment in the IBM Data Science Specialization
Csv of Ottawa geospatial data containing postal codes, city, neighbourhood, latitude and longitude - https://github.com/ccnixon/postalcodes/blob/master/CanadianPostalCodes.csv
Geospatial coordinates of Toronto containing latitude and longitude - from a previous assessment in the IBM Data Science Specialization
Foursquare API location data for venues in Toronto and Ottawa – including venue latitude, longitude, category and name - https://api.foursquare.com/v2/ . Limit 100. Radius 500.

Question 2 data:

Csv of Toronto location data containing postal codes, boroughs and neighbourhoods - from a previous assessment in the IBM Data Science Specialization
Geospatial coordinates or Toronto containing latitude and longitude - from a previous assessment in the IBM Data Science Specialization
Foursquare API location data for italian restaurants in Toronto – including name, address, latitude, longitude, distance, postal code, city, state and neighbourhood - https://api.foursquare.com/v2/ . Limit 100. Radius 10000.

b. Data cleaning

Question 1 data cleaning: within the data sources section of this report you can see the Toronto data is separated across two csv files. The process of data cleaning involved joining these sources and inputting them into the same pandas dataframe. After which all boroughs that contained the value "Not assigned" were dropped from the dataframe and filtered to the Downtown Toronto borough. The venue data from the Foursquare API was filtered to only include venue name, category, latitude, longitude in Toronto and Ottawa respectively. The Ottawa data required no cleaning.

Question 2 data cleaning: as I was using the same Toronto data from question 1 no additional cleaning of the csv files was required. It is important to note that in this case the Toronto data was not limited to the Downtown borough. The Foursquare API data for this question was filtered to Toronto with an additional search query equal to italian.

c. Feature selection

Question 1 features: venues in Downtown Toronto and Ottawa respectively from the Foursquare API

Question 2 features: Italian restaurants in Toronto from the Foursquare API

Methodology

a. Exploratory data analysis

Question 1 – exploratory data analysis

Objective: to compare the neighbourhoods of Downtown Toronto and Ottawa and determine how similar or dissimilar they are. In this section I will define the methodology used for exploratory data analysis related to question 1.

Toronto dataset:

1. Merging dataframes and data cleaning. The Toronto dataset is a merger of two data sources containing different metadata. The first dataset lists the postal codes, boroughs and neighbourhoods of Toronto, whilst the second contains geospatial information – latitude and longitude. In order to visualise the data on a folium map it was necessary to merge these sources into a single dataframe containing all the metadata listed above. After exploring the dataset I discovered that some boroughs contained the equivalent of blank values – "Not assigned", these were then dropped from the dataframe as they cannot be used in my analysis.
2. Initial exploration of the dataset involved investigating the dimensionality (.shape), descriptive statistics (.describe) and the data types (.dtypes) that exist in the dataset.
3. Validating the geospatial data. With the resulting merged dataframe it was now necessary to validate the accuracy of the location data. This was done visually using the folium package, defining the pop-up label to contain the related neighbourhood and borough.
4. Simplifying the dataset. In order to demonstrate additional skills I filtered the Toronto dataset to only include metadata associated with the borough of "Downtown Toronto".
5. Exploring the first neighbourhood in the filtered Toronto dataframe. Using the first neighbourhood listed in the new dataframe "Queen's Park, Ontario Provincial Government", I explored the corresponding geospatial information, and the top 100 venues within a 500 metre radius using the Foursquare API.
    a. Borrowing the get_category function from the Foursquare API I categorized the resulting API request metadata (json) and committed the results to a new dataframe.
6. Exploring all neighbourhoods in Downtown Toronto. I applied the method used for exploring the first neighbourhood in Downtown Toronto to all neighbourhoods be defining a getNearbyVenues function. This function looped through all neighbourhoods in Downtown Toronto and applied the same criteria – top 100 venues within a 500 metre radius via the Foursquare API, borrowing the get_category function. The venues were then grouped by neighbourhood and committed to a new dataframe – with the number of unique categories stated.
7. Analysing each neighbourhood.
    a. One hot encoding. As the venue data returned by the Foursquare API is categorical there is a need to convert it to numerical for further analysis. In order to do this I used one hot encoding to assign dummy values to each category. After which I grouped the data by neighbourhood and took the mean frequency of occurrence in each category.
    b. Top 10 most common venues. I printed the top 5 most common venues in each neighbourhood and wrote a function that sorted the top 10 venues in descending order. The sorted venues were then committed to a new dataframe, tagged to their related neighbourhood.

Ottawa dataset:

1. Loading the dataset and data cleaning. The Ottawa dataset contained all the required metadata for plotting folium maps – postal codes, city, neighbourhood, latitude and longitude. The dataset also did not contain any blank values or equivalents therefore no cleaning was necessary.
2. Initial exploration of the dataset involved investigating the dimensionality (.shape), descriptive statistics (.describe) and the data types (.dtypes) that exist in the dataset.
3. Validating the geospatial data. With the resulting dataframe it was now necessary to validate the accuracy of the location data. This was done using the folium package, defining the pop-up label to contain the related neighbourhood.
4. Simplifying the dataset. In order to demonstrate additional skills I filtered the Ottawa dataset to only include metadata associated with the neighbourhood “Downtown”.
5. Exploring the first neighbourhood in the filtered Ottawa dataframe. Using the first neighbourhood listed in the new dataframe “Downtown”, I explored the corresponding geospatial information, and the top 100 venues within a 500 metre radius using the Foursquare API.
    a. Borrowing the get_category function from the Foursquare API I categorized the resulting API request metadata (json) and committed the results to a new dataframe.
6. Exploring all neighbourhoods in Ottawa. I applied the method used for exploring the first neighbourhood in Ottawa to all neighbourhoods be defining a getNearbyVenues function. This function looped through all neighbourhoods in Downtown Toronto and applied the same criteria – top 100 venues within a 500 metre radius via the Foursquare API, borrowing the get_category function. The venues were then grouped by neighbourhood and committed to a new dataframe – with the number of unique categories stated.
7. Analysing each neighbourhood.
    a. One hot encoding. As the venue data returned by the Foursquare API is categorical there is a need to convert it to numerical for further analysis. In order to do this I used one hot encoding to assign dummy values to each category. After which I grouped the data by neighbourhood and took the mean frequency of occurrence in each category.
    b. Top 10 most common venues. I printed the top 5 most common venues in each neighbourhood and wrote a function that sorted the top 10 venues in descending order. The sorted venues were then committed to a new dataframe, tagged to their related neighbourhood.

Question 2 – exploratory data analysis

Objective: A restaurant owner is looking to open a new Italian restaurant in Toronto, the objective is to recommend the best area in which a new restaurant could be located.. In this section I will define the methodology used for exploratory data analysis related to question 1.

1. To prevent duplication the data used for this question is the same underlying Toronto dataset discussed in question 1. Therefore, from an exploratory data analysis point of view this was already completed – please see question 1 exploratory data analysis for the Toronto dataset above – points 1,2, and 3.
2. Geospatial data. Unlike question 1 the data used in this question covers the all of Toronto and is not isolated to a particular neighbourhood. Using geocoder I converted the city of Toronto into latitude and longitude coordinates.
3. Exploring Italian restaurants in Toronto. Using the latitude and longitude coordinate results from geocoder I requested a search query from the Foursquare API, specifying the search query

"Italian" and radius of 10,000 metres. This enabled the identification of all Italian restaurants within the city of Toronto.

4. Data cleaning. The results from the Foursquare API contained a vast amount of irrelevant metadata so after converting the json output to a new dataframe I dropped these columns.
5. Venue category. I defined and looped the get category function to the entire dataframe enabling the extraction of the venue category, committing the new found category to the dataframe.
6. Visualization. The intention of this question is to identify the spatial location of all pre-existing Italian restaurants in Toronto, enabling the restaurant owner to make an informed decision on where to build his new restaurant. The spatial distribution is highly important from a competition point of view as an area highly saturated in Italian cuisine will prove detrimental to their business. Therefore, the owner is looking for an area that has none/few Italian restaurants at present. To satisfy this requirement I visualised the current Italian restaurants on a folium map – informing the owner of current spatial distribution.
7. Choosing the location of a new Italian restaurant. Now that I have identified the best areas for a new Italian restaurant the owner can now focus on building his venue in one of these areas. The specific address of the new Italian restaurant is beyond the scope of this project as it depends on various external factors – such as availability of building for sale etc.

b. MACHINE LEARNING

Question 1 – machine learning

Objective: to compare the neighbourhoods of Downtown Toronto and Ottawa and determine how similar or dissimilar they are. In this section I will define the methodology used for machine learning related to question 1.

The most appropriate machine learning technique for the scenario detailed in this report is k-means clustering as by definition k-means clusters data based on its similarity. The objective of question 1 is to determine the similarity or dissimilarity between the neighbourhoods of Downtown Toronto and Ottawa so k-means clustering is a natural fit. The methodology below describes how I used venues to group neighbourhoods into categories.

Toronto dataset:

1. Following on from point 7 in the exploratory data analysis section, in which I identified the top 10 most common venues in each neighbourhood. I determined the optimum value of k using the elbow method and ran k-means clustering on the neighbourhoods in Downtown Toronto – resulting in clusters of neighbourhoods based on their venues. I then generated a new dataframe containing the new cluster labels and top 10 most common venues in each neighbourhood.
2. Visualising the clusters. Using the newly generated dataframe containing the cluster labels I visualised the clusters on a folium map to determine their spatial distribution.
3. Examining the clusters. In order to identify the venue categories that distinguish each cluster I examined each cluster group separately within my jupyter notebook.

Ottawa dataset:

1. Following on from point 7 in the exploratory data analysis section, in which I identified the top 10 most common venues in each neighbourhood. I determined the optimal value of k using the elbow method and ran k-means clustering on the neighbourhoods in Ottawa – resulting in clusters of

neighbourhoods based on their venues. I then generated a new dataframe containing the new cluster labels and top 10 most common venues in each neighbourhood.

2. Visualising the clusters. Using the newly generated dataframe containing the cluster labels I visualised the clusters on a folium map to determine their spatial distribution.
3. Examining the clusters. In order to identify the venue categories that distinguish each cluster I examined each cluster group separately within my jupyter notebook.

C. RESULTS

Question 1 – results

Objective: to compare the neighbourhoods of Downtown Toronto and Ottawa and determine how similar or dissimilar they are. In this section I will detail the results of my analysis – for the code please see my jupyter notebook.

Toronto dataset:

1. Merged and cleaned dataset.

Loading the Toronto and Ottawa data into two seperate pandas dataframes:

```
# Loading toronto wiki table of postcodes
df1 = pd.read_csv('postal_codes_of_canada.csv')

# Dropping the Borough rows that contain the value "Not assigned"
df1 = df1[df1.Borough != "Not assigned"]

# Loading csv file of cooordinates
coords_df = pd.read_csv("Geospatial_Coordinates.csv")
coords_df.head()

#merging the two dataframes into a single df_toronto - using the Postal Code.
df_toronto = pd.merge(df1,coords_df,on="Postal Code")
# filtering the df to only include boroughs that contain the word "toronto".
df_toronto = df_toronto[df_toronto['Borough'].str.contains("Toronto")]

#renaming neighborhood column
df_toronto.rename(columns={'Neighbourhood': 'Neighborhood'}, inplace=True)
df_toronto.head(3)
```

	Postal Code	Borough	Neighborhood	Latitude	Longitude
15	M7Y	East Toronto	Business reply mail Processing Centre, South C...	43.662744	-79.321558
17	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494
18	M6S	West Toronto	Runnymede, Swansea	43.651571	-79.484450

2. Dimensionality, descriptive statistics and data types.

```
df_toronto.shape
```

```
(39, 5)
```

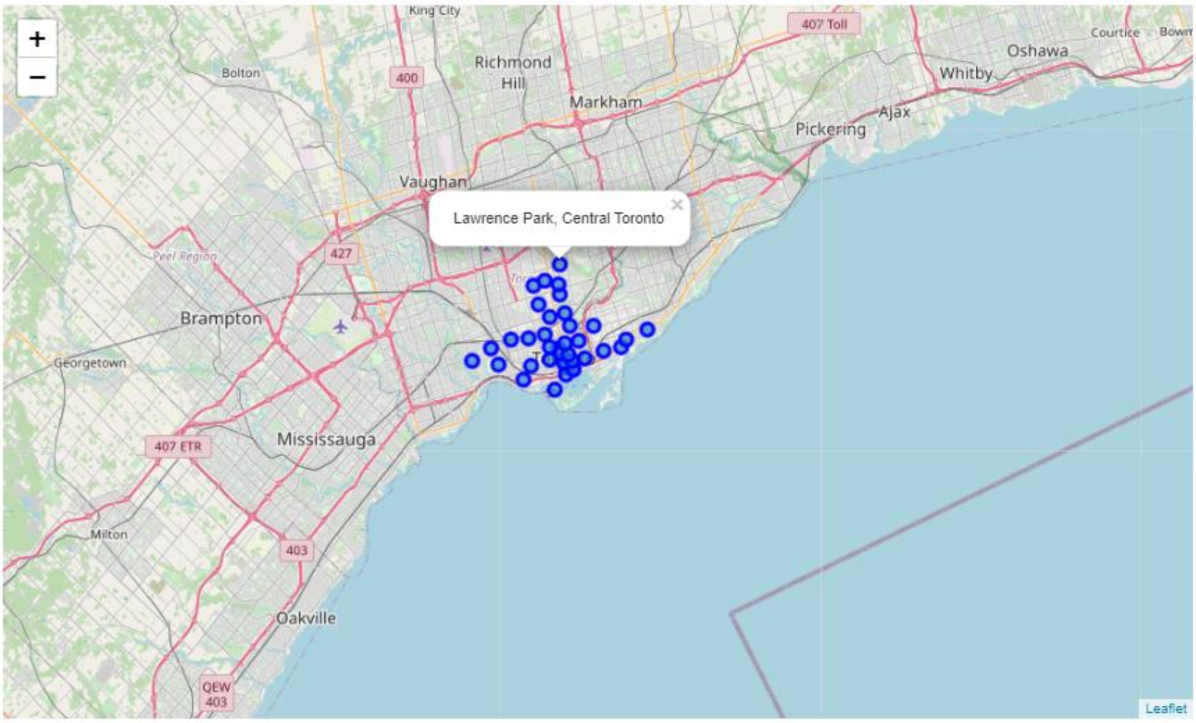
```
df_toronto.describe()
```

	Latitude	Longitude
count	39.000000	39.000000
mean	43.667135	-79.389873
std	0.023478	0.037451
min	43.628947	-79.484450
25%	43.649765	-79.405678
50%	43.662301	-79.387383
75%	43.677957	-79.376474
max	43.728020	-79.293031

```
df_toronto.dtypes
```

```
Postal Code     object
Borough         object
Neighborhood    object
Latitude       float64
Longitude      float64
dtype: object
```

3. Validating the geospatial data.



4. Simplifying the dataset.

Focusing on the Toronto dataset

Simplifying the Toronto data to cluster only the neighborhoods in Downtown Toronto

```
toronto_downtown_data = df_toronto[df_toronto["Borough"]=="Downtown Toronto"].reset_index(drop=True)
toronto_downtown_data.head(10)
```

	Postal Code	Borough	Neighborhood	Latitude	Longitude
0	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494
1	M6G	Downtown Toronto	Christie	43.669542	-79.422564
2	M5X	Downtown Toronto	First Canadian Place, Underground city	43.648429	-79.382280
3	M5W	Downtown Toronto	Stn A PO Boxes	43.646435	-79.374846
4	M5V	Downtown Toronto	CN Tower, King and Spadina, Railway Lands, Har...	43.628947	-79.394420
5	M5T	Downtown Toronto	Kensington Market, Chinatown, Grange Park	43.653206	-79.400049
6	M5S	Downtown Toronto	University of Toronto, Harbord	43.662696	-79.400049
7	M5L	Downtown Toronto	Commerce Court, Victoria Hotel	43.648198	-79.379817
8	M5K	Downtown Toronto	Toronto Dominion Centre, Design Exchange	43.647177	-79.381576
9	M5J	Downtown Toronto	Harbourfront East, Union Station, Toronto Islands	43.640816	-79.381752

5. Exploring the first neighbourhood and the top 100 venues in a radius of 500m, with resulting dataframe.

Exploring the first neighborhood in the Toronto dataframe

```
toronto_downtown_data.loc[0,"Neighborhood"]
```

```
"Queen's Park, Ontario Provincial Government"
```

```
neighborhood_latitude = toronto_downtown_data.loc[0, 'Latitude'] # neighborhood latitude value
neighborhood_longitude = toronto_downtown_data.loc[0, 'Longitude'] # neighborhood longitude value

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

print('Latitude and longitude values of {} are {}, {}.'.format(neighborhood_name,
                                                               neighborhood_latitude,
                                                               neighborhood_longitude))
```

```
Latitude and longitude values of Queen's Park, Ontario Provincial Government are 43.6623015, -79.3894938.
```

Define Foursquare credentials

```
CLIENT_ID = [redacted]' # your Foursquare ID
CLIENT_SECRET = [redacted] # your Foursquare Secret
VERSION = '20180605' # Foursquare API version
LIMIT = 100 # A default Foursquare API limit value

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: XBAJLFUBKHRRDCJOZ3PGS4R2EOIDLVOGMCX5PKJJK3HXXQHN
CLIENT_SECRET:XCXXBLEAC1JRK1TA2URBNDK2INZO4VT5RIZB0EUUBU2ALIDX
```

Exploring the top 100 venues that are in Queen's Park within a radius of 500 meters.

```
LIMIT = 100
radius = 500
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
```

	name	categories	lat	lng
0	Queen's Park	Park	43.663946	-79.392180
1	Mercatto	Italian Restaurant	43.660391	-79.387664
2	NEO COFFEE BAR	Coffee Shop	43.660130	-79.385830
3	Central YMCA	Distribution Center	43.663083	-79.385025
4	T-Swirl Crepe	Creperie	43.663452	-79.384125

```
print('{} venues were returned by Foursquare.'.format(nearby_venues.shape[0]))
```

```
33 venues were returned by Foursquare.
```

6. Exploring all neighbourhoods in Downtown Toronto, grouping venues by neighbourhood

Grouping venues in Downtown Toronto by neighborhood

```
toronto_downtown_venues.groupby('Neighborhood').count()
```

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Berczy Park	56	56	56	56	56	56
CN Tower, King and Spadina, Railway Lands, Harbourfront West, Bathurst Quay, South Niagara, Island airport	17	17	17	17	17	17
Central Bay Street	63	63	63	63	63	63
Christie	16	16	16	16	16	16
Church and Wellesley	77	77	77	77	77	77
Commerce Court, Victoria Hotel	100	100	100	100	100	100
First Canadian Place, Underground city	100	100	100	100	100	100
Garden District, Ryerson	100	100	100	100	100	100
Harbourfront East, Union Station, Toronto Islands	100	100	100	100	100	100
Kensington Market, Chinatown, Grange Park	63	63	63	63	63	63
Queen's Park, Ontario Provincial Government	33	33	33	33	33	33
Regent Park, Harbourfront	43	43	43	43	43	43
Richmond, Adelaide, King	92	92	92	92	92	92
Rosedale	4	4	4	4	4	4
St. James Town	80	80	80	80	80	80
St. James Town, Cabbagetown	45	45	45	45	45	45
Stn A PO Boxes	98	98	98	98	98	98
Toronto Dominion Centre, Design Exchange	100	100	100	100	100	100
University of Toronto, Harbord	31	31	31	31	31	31

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

```
There are 206 uniques categories.
```

7. Analysing each neighbourhood, with the top 10 most common venues in each neighbourhood detailed.

Defining a new dataframe and displaying the top 10 venues for each neighborhood within Downtown Toronto

```
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
downtown_toronto_neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
downtown_toronto_neighborhoods_venues_sorted['Neighborhood'] = downtown_toronto_grouped['Neighborhood']

for ind in np.arange(downtown_toronto_grouped.shape[0]):
    downtown_toronto_neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(downtown_toronto_grouped.iloc[ind, :], nu

downtown_toronto_neighborhoods_venues_sorted.head()
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Berczy Park	Coffee Shop	Cocktail Bar	Cheese Shop	Farmers Market	Seafood Restaurant	Bakery	Beer Bar	Restaurant	Eastern European Restaurant	Hotel
1	CN Tower, King and Spadina, Railway Lands, Har...	Airport Service	Airport Lounge	Airport Terminal	Boat or Ferry	Sculpture Garden	Plane	Boutique	Rental Car Location	Harbor / Marina	Coffee Shop
2	Central Bay Street	Coffee Shop	Sandwich Place	Italian Restaurant	Café	Salad Place	Bubble Tea Shop	Burger Joint	Wine Bar	Miscellaneous Shop	Japanese Restaurant
3	Christie	Grocery Store	Café	Park	Italian Restaurant	Athletics & Sports	Candy Store	Restaurant	Baby Store	Nightclub	Coffee Shop
4	Church and Wellesley	Coffee Shop	Japanese Restaurant	Sushi Restaurant	Restaurant	Gay Bar	Mediterranean Restaurant	Men's Store	Burger Joint	Hotel	Pub

Ottawa dataset:

1. Loading the Ottawa dataset

```
# Loading otawa geospatial data
df_ottawa = pd.read_csv('ottawa_geospatial.csv')

#renaming neighborhood column
df_ottawa.rename(columns={'Neighbourhood': 'Neighborhood'}, inplace=True)
df_ottawa.head(3)
```

	Postal Code	City	Neighborhood	Latitude	Longitude
0	K2A	Ottawa	Highland Park, McKellar Park,Westboro, Glabar ...	45.195828	-75.823060
1	K4A	Ottawa	Fallingbrook	45.449549	-75.476860
2	K1B	Ottawa	Blackburn Hamlet, Pine View, Sheffield Glen	45.416035	-75.621899

2. Dimensionality, descriptive statistics and data types.

```
df_ottawa.shape
```

```
(40, 5)
```

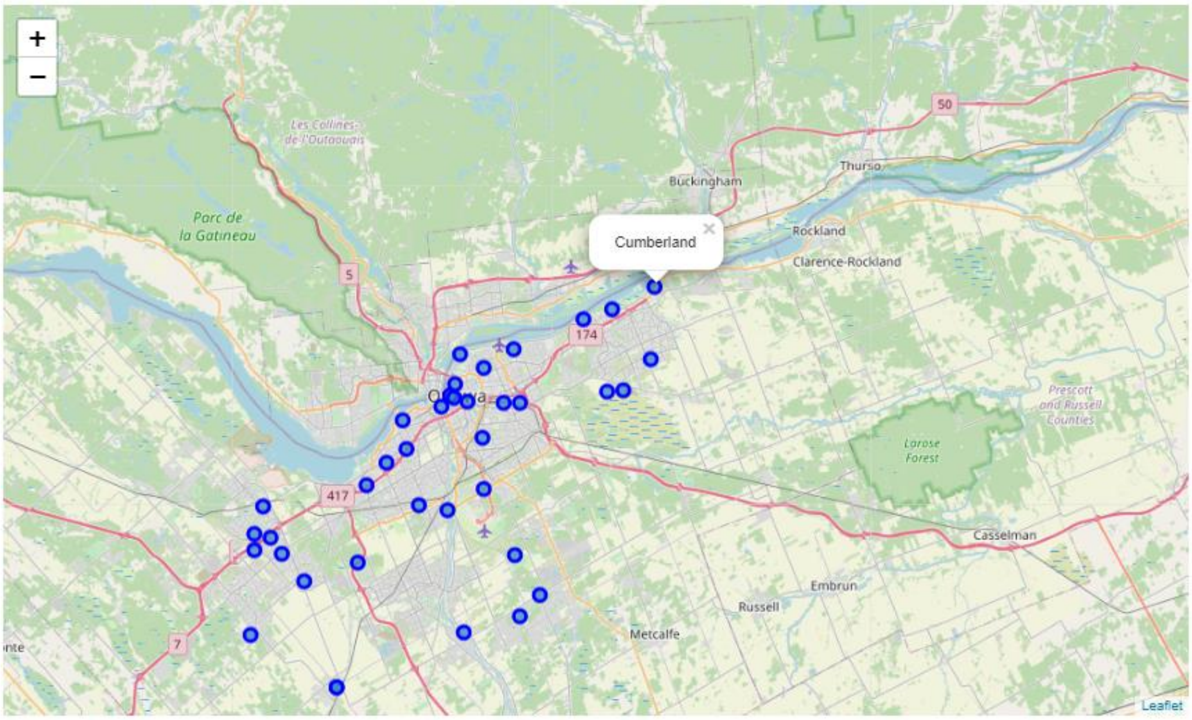
```
df_ottawa.describe()
```

	Latitude	Longitude
count	40.000000	40.000000
mean	45.334415	-75.750297
std	0.193601	0.244767
min	44.249605	-76.955967
25%	45.298361	-75.823060
50%	45.360518	-75.696435
75%	45.422508	-75.627695
max	45.505467	-75.472889

```
df_ottawa.dtypes
```

```
Postal Code      object
City             object
Neighborhood     object
Latitude        float64
Longitude       float64
dtype: object
```

3. Validating the geospatial data.



4. Exploring the first neighbourhood and the top 100 venues in a radius of 500m, with resulting dataframe.

```
ott_venues = results2['response']['groups'][0]['items']

ott_nearby_venues = json_normalize(ott_venues) # flatten JSON

# filter columns
ott_filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 've
ott_nearby_venues =ott_nearby_venues.loc[:, ott_filtered_columns]

# filter the category for each row
ott_nearby_venues['venue.categories'] = ott_nearby_venues.apply(get_category_type,

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

ott_nearby_venues.head()
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages/ipykernel_launcher.p
is deprecated, use pandas.json_normalize instead
  This is separate from the ipykernel package so we can avoid doing imports until
```

	name	categories	lat	lng
0	Alt Hotel Ottawa	Hotel	45.419973	-75.698948
1	Cafe Deluxe	Café	45.421744	-75.696310
2	Riviera	Modern European Restaurant	45.423165	-75.696205
3	Sheraton Ottawa Hotel	Hotel	45.420846	-75.697712
4	Northern Lights Sound and Light Show	Plaza	45.423856	-75.698520

5. Exploring all neighbourhoods in Downtown Toronto, grouping venues by neighbourhood

```
ottawa_venues.groupby('Neighborhood').count()
```

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Alta Vista, Billings Bridge	2	2	2	2	2	2
Barrhaven	29	29	29	29	29	29
Beacon Hill, Cyrville, Carson Grove	1	1	1	1	1	1
Beaverbrook, South March	15	15	15	15	15	15
Bells Corners, Arlington Woods, Redwood, Qualicum, Crystal Beach	4	4	4	4	4	4
Blackburn Hamlet, Pine View, Sheffield Glen	2	2	2	2	2	2
Blossom Park, Greenboro, Leitrim, Findlay Creek	1	1	1	1	1	1
Bridlewood	3	3	3	3	3	3
Britannia, Whitehaven, Bayshore, Pinecrest	9	9	9	9	9	9
Centrepointe, Meadowlands, City View, Craig Henry, Tangelwood, Grenfell Glen, Davidson Heights	7	7	7	7	7	7
Centretown	78	78	78	78	78	78
Civic Hospital, Island Park, Hintonburg, Mechanicsville, Champlain Park	3	3	3	3	3	3
Dalhousie Ward	33	33	33	33	33	33
Downtown	96	96	96	96	96	96
Fallingbrook	6	6	6	6	6	6
Fisher Heights, Parkwood Hills, Borden Farm, Pine Glen	3	3	3	3	3	3
Heron Gate, Heron Park, Riverside Park, Hunt Club, Riverside South, YOW	2	2	2	2	2	2
Katimavik-Hazeldean, Glen Cairn	5	5	5	5	5	5
Lower Town, Byward Market, Sandy Hill, University of Ottawa	100	100	100	100	100	100
Marchwood	6	6	6	6	6	6
Navan	2	2	2	2	2	2
North March	25	25	25	25	25	25
Orleans	1	1	1	1	1	1

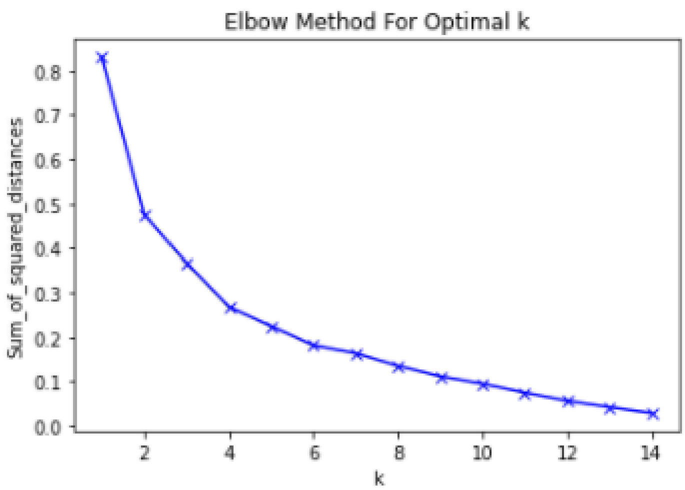
6. Analysing each neighbourhood, with the top 10 most common venues in each neighbourhood detailed.

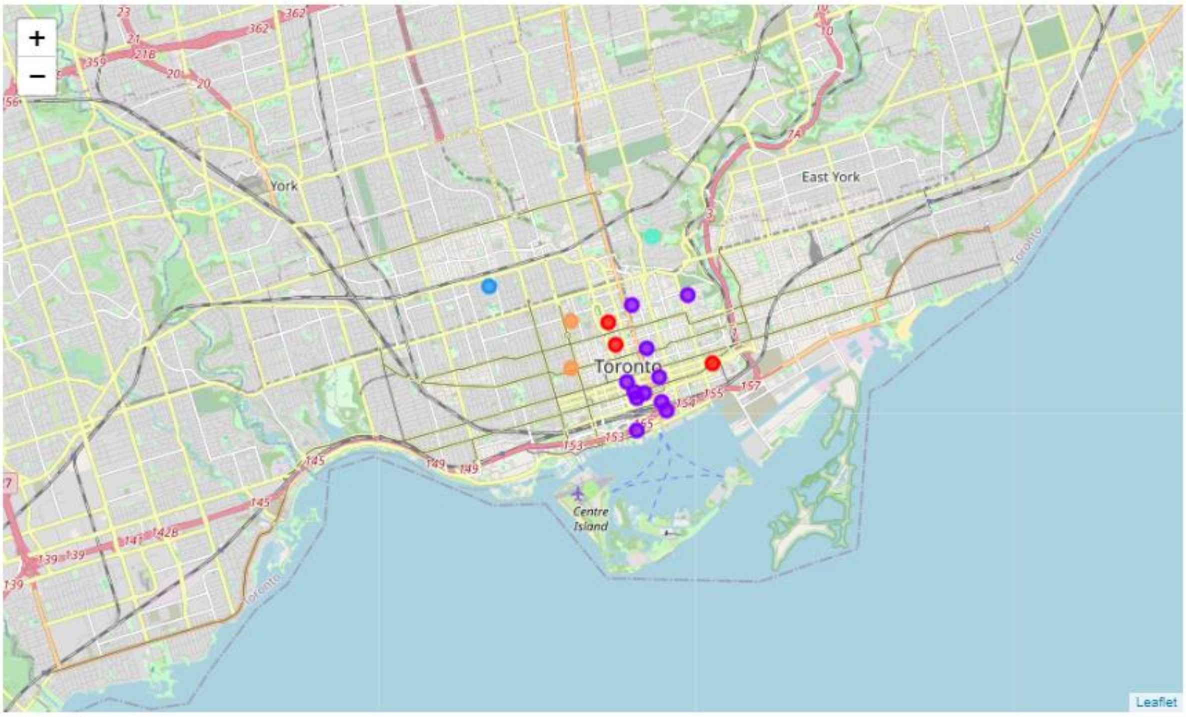
```
num_top_venues = 10

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
ottawa_neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
ottawa_neighborhoods_venues_sorted['Neighborhood'] = ottawa_grouped['Neighborhood']

for ind in np.arange(ottawa_grouped.shape[0]):
    ottawa_neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(ottawa_grouped.iloc[ind, :], num_top_venues)

ottawa_neighborhoods_venues_sorted.head()
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Alta Vista, Billings Bridge	Furniture / Home Store	Grocery Store	Garden Center	Frozen Yogurt Shop	Fried Chicken Joint	French Restaurant	Food Truck	Food & Drink Shop	Fast Food Restaurant	Farmers Market
1	Barrhaven	Coffee Shop	Chinese Restaurant	Restaurant	Bank	Grocery Store	Shopping Mall	Café	Sandwich Place	Fast Food Restaurant	Luggage Store
2	Beacon Hill, Cyrville, Carson Grove	Theme Park	Yoga Studio	Garden Center	Frozen Yogurt Shop	Fried Chicken Joint	French Restaurant	Food Truck	Food & Drink Shop	Fast Food Restaurant	Farmers Market
3	Beaverbrook, South March	Gym / Fitness Center	Indian Restaurant	Bank	Dance Studio	Coffee Shop	Sandwich Place	Café	Mexican Restaurant	Breakfast Spot	Sushi Restaurant
4	Bells Corners, Arlington Woods, Redwood, Quali...	Pharmacy	History Museum	Train Station	Grocery Store	Yoga Studio	Dive Bar	French Restaurant	Food Truck	Food & Drink Shop	Fast Food Restaurant

Question 1 – machine learning results.

Toronto dataset – k means clustering.





Individual cluster analysis:

Examining the clusters in Downtown Toronto

```
# cluster 1
downtown_toronto_merged.loc[downtown_toronto_merged['Cluster Labels'] == 0, downtown_toronto_merged.columns[[1] + list(range(5, dow
```

	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 Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Downtown Toronto	0	Coffee Shop	Sushi Restaurant	Yoga Studio	Diner	Smoothie Shop	Italian Restaurant	Beer Bar	Music Venue	Sandwich Place	Distribution Center
11	Downtown Toronto	0	Coffee Shop	Sandwich Place	Italian Restaurant	Café	Salad Place	Bubble Tea Shop	Burger Joint	Wine Bar	Miscellaneous Shop	Japanese Restaurant
15	Downtown Toronto	0	Coffee Shop	Bakery	Park	Breakfast Spot	Theater	Café	Pub	Farmers Market	Event Space	Shoe Store

```
# cluster 2
downtown_toronto_merged.loc[downtown_toronto_merged['Cluster Labels'] == 1, downtown_toronto_merged.columns[[1] + list(range(5, do
```

	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 Most Common Venue	9th Most Common Venue	10th Most Common Venue
2	Downtown Toronto	1	Coffee Shop	Café	Hotel	Restaurant	Japanese Restaurant	Gym	Deli / Bodega	American Restaurant	Asian Restaurant	Salad Place
3	Downtown Toronto	1	Coffee Shop	Seafood Restaurant	Restaurant	Japanese Restaurant	Italian Restaurant	Cocktail Bar	Hotel	Beer Bar	Café	Cheese Shop
7	Downtown Toronto	1	Coffee Shop	Restaurant	Café	Hotel	Gym	Italian Restaurant	American Restaurant	Seafood Restaurant	Japanese Restaurant	Deli / Bodega
8	Downtown Toronto	1	Coffee Shop	Hotel	Café	Seafood Restaurant	American Restaurant	Salad Place	Italian Restaurant	Restaurant	Japanese Restaurant	Gastropub
9	Downtown Toronto	1	Coffee Shop	Aquarium	Hotel	Café	Brewery	Scenic Lookout	Fried Chicken Joint	Restaurant	Sporting Goods Shop	Italian Restaurant
10	Downtown Toronto	1	Coffee Shop	Café	Restaurant	Deli / Bodega	Clothing Store	Thai Restaurant	Bakery	Gym	Salad Place	Bookstore
12	Downtown Toronto	1	Coffee Shop	Cocktail Bar	Cheese Shop	Farmers Market	Seafood Restaurant	Bakery	Beer Bar	Restaurant	Eastern European Restaurant	Hotel
13	Downtown Toronto	1	Coffee Shop	Café	Gastropub	Cocktail Bar	American Restaurant	Department Store	Moroccan Restaurant	Lingerie Store	Cosmetics Shop	Clothing Store
14	Downtown Toronto	1	Clothing Store	Coffee Shop	Bubble Tea Shop	Middle Eastern Restaurant	Café	Japanese Restaurant	Italian Restaurant	Cosmetics Shop	Ramen Restaurant	Movie Theater
16	Downtown Toronto	1	Coffee Shop	Japanese Restaurant	Sushi Restaurant	Restaurant	Gay Bar	Mediterranean Restaurant	Men's Store	Burger Joint	Hotel	Pub
17	Downtown Toronto	1	Coffee Shop	Café	Bakery	Chinese Restaurant	Pub	Park	Restaurant	Italian Restaurant	Pizza Place	Liquor Store

```
# cluster 3
downtown_toronto_merged.loc[downtown_toronto_merged['Cluster Labels'] == 2, downtown_toronto_merged.columns[[1] + list(range(5, down
```

	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 Most Common Venue	9th Most Common Venue	10th Most Common Venue
1	Downtown Toronto	2	Grocery Store	Café	Park	Italian Restaurant	Athletics & Sports	Candy Store	Restaurant	Baby Store	Nightclub	Coffee Shop

```
# cluster 4
downtown_toronto_merged.loc[downtown_toronto_merged['Cluster Labels'] == 3, downtown_toronto_merged.columns[[1] + list(range(5, down
```

	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 Most Common Venue	9th Most Common Venue	10th Most Common Venue
18	Downtown Toronto	3	Park	Trail	Playground	Creperie	Doner Restaurant	Dog Run	Distribution Center	Discount Store	Diner	Dessert Shop

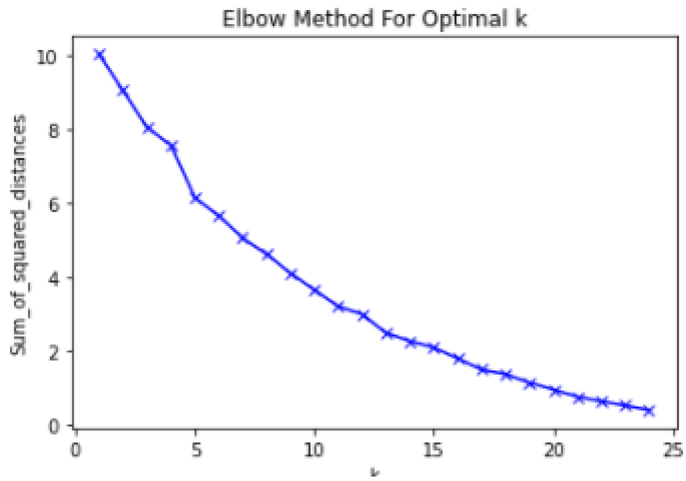
```
# cluster 5
downtown_toronto_merged.loc[downtown_toronto_merged['Cluster Labels'] == 4, downtown_toronto_merged.columns[[1] + list(range(5, down
```

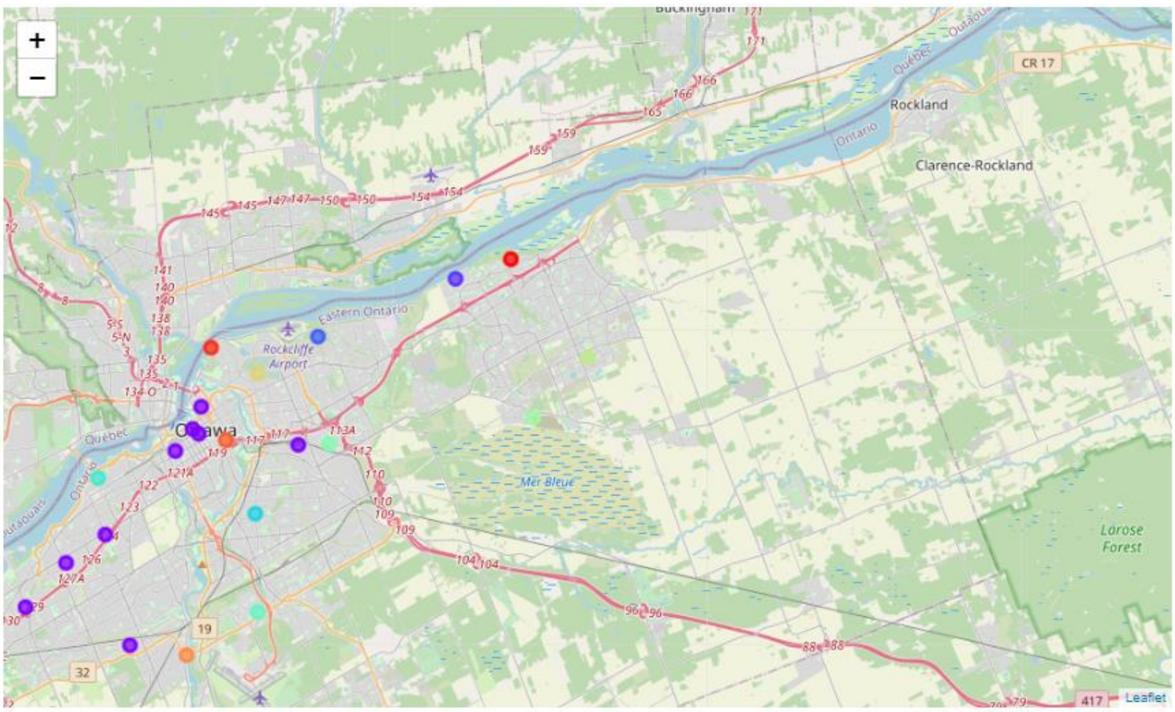
	Borough	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
4	Downtown Toronto	4	Airport Service	Airport Lounge	Airport Terminal	Boat or Ferry	Sculpture Garden	Plane	Boutique	Rental Car Location	Harbor / Marina	Coffee Shop

```
# cluster 6
downtown_toronto_merged.loc[downtown_toronto_merged['Cluster Labels'] == 5, downtown_toronto_merged.columns[[1] + list(range(5, down
```

	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 Most Common Venue	9th Most Common Venue	10th Most Common Venue
5	Downtown Toronto	5	Café	Vegetarian / Vegan Restaurant	Coffee Shop	Mexican Restaurant	Bakery	Bar	Vietnamese Restaurant	Park	Grocery Store	Gaming Cafe
6	Downtown Toronto	5	Café	Bookstore	Bar	Japanese Restaurant	Bakery	Yoga Studio	Italian Restaurant	Beer Bar	College Gym	Sandwich Place

Ottawa dataset – k means clustering





Individual cluster analysis

Examining the Ottawa clusters

```
# cluster 1
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 0, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
8	Ottawa	0.0	Clothing Store	Trail	Yoga Studio	Electronics Store	Frozen Yogurt Shop	Fried Chicken Joint	French Restaurant	Food Truck	Food & Drink Shop	Fast Food Restaurant

```
# cluster 2
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 1, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
3	Ottawa	1.0	Coffee Shop	Department Store	Storage Facility	Gym	Bus Station	Bookstore	Pizza Place	Health Food Store	Dive Bar	French Restaurant
10	Ottawa	1.0	Clothing Store	Sporting Goods Shop	Gym / Fitness Center	Shoe Store	Gas Station	Pet Store	Bridal Shop	Food & Drink Shop	Liquor Store	Big Box Store
11	Ottawa	1.0	Hardware Store	Food Truck	Park	Auto Garage	Clothing Store	Furniture / Home Store	Liquor Store	Electronics Store	Fried Chicken Joint	French Restaurant
13	Ottawa	1.0	Pharmacy	History Museum	Train Station	Grocery Store	Yoga Studio	Dive Bar	French Restaurant	Food Truck	Food & Drink Shop	Fast Food Restaurant
15	Ottawa	1.0	Coffee Shop	Chinese Restaurant	Restaurant	Bank	Grocery Store	Shopping Mall	Café	Sandwich Place	Fast Food Restaurant	Luggage Store
17	Ottawa	1.0	Gym / Fitness Center	Indian Restaurant	Bank	Dance Studio	Coffee Shop	Sandwich Place	Café	Mexican Restaurant	Breakfast Spot	Sushi Restaurant
19	Ottawa	1.0	Pool	Dance Studio	Supermarket	Chinese Restaurant	Optical Shop	Deli / Bodega	Department Store	Dessert Shop	Diner	Dive Bar
23	Ottawa	1.0	Pub	New American Restaurant	Café	Coffee Shop	Dessert Shop	BBQ Joint	Vietnamese Restaurant	Hotel	Ice Cream Shop	Furniture / Home Store
						Sandwich		Japanese		Sushi		

```
# cluster 3
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 2, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
5	Ottawa	2.0	Construction & Landscaping	Electronics Store	Frozen Yogurt Shop	Fried Chicken Joint	French Restaurant	Food Truck	Food & Drink Shop	Fast Food Restaurant	Farmers Market	Dive Bar

```
# cluster 4
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 3, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
16	Ottawa	3.0	Boat or Ferry	Yoga Studio	Gas Station	Furniture / Home Store	Frozen Yogurt Shop	Fried Chicken Joint	French Restaurant	Food Truck	Food & Drink Shop	Fast Food Restaurant

```
# cluster 5
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 4, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
31	Ottawa	4.0	Campground	Yoga Studio	Farmers Market	Frozen Yogurt Shop	Fried Chicken Joint	French Restaurant	Food Truck	Food & Drink Shop	Fast Food Restaurant	Electronics Store

```
# cluster 6
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 5, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
14	Ottawa	5.0	Theme Park	Yoga Studio	Garden Center	Frozen Yogurt Shop	Fried Chicken Joint	French Restaurant	Food Truck	Food & Drink Shop	Fast Food Restaurant	Farmers Market

```
# cluster 7
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 6, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
12	Ottawa	6.0	Furniture / Home Store	Grocery Store	Garden Center	Frozen Yogurt Shop	Fried Chicken Joint	French Restaurant	Food Truck	Food & Drink Shop	Fast Food Restaurant	Farmers Market

```
# cluster 8
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 7, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
38	Ottawa	7.0	Sculpture Garden	Playground	Yoga Studio	Dive Bar	Fried Chicken Joint	French Restaurant	Food Truck	Food & Drink Shop	Fast Food Restaurant	Farmers Market

```
# cluster 9
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 8, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
33	Ottawa	8.0	Bus Station	Park	Yoga Studio	Farmers Market	Frozen Yogurt Shop	Fried Chicken Joint	French Restaurant	Food Truck	Food & Drink Shop	Fast Food Restaurant

```
# cluster 10
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 9, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
2	Ottawa	9.0	Business Service	Shop & Service	Yoga Studio	Cosmetics Shop	Frozen Yogurt Shop	Fried Chicken Joint	French Restaurant	Food Truck	Food & Drink Shop	Fast Food Restaurant

```
# cluster 11
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 10, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
4	Ottawa	10.0	Waste Facility	Garden Center	Yoga Studio	Electronics Store	Frozen Yogurt Shop	Fried Chicken Joint	French Restaurant	Food Truck	Food & Drink Shop	Fast Food Restaurant

```
# cluster 12
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 11, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
1	Ottawa	11.0	Construction & Landscaping	Home Service	Pharmacy	Boutique	Sports Club	Farmers Market	Frozen Yogurt Shop	Fried Chicken Joint	French Restaurant	Food Truck
30	Ottawa	11.0	Construction & Landscaping	Home Service	Business Service	Grocery Store	Yoga Studio	Farmers Market	Fried Chicken Joint	French Restaurant	Food Truck	Food & Drink Shop

```python
# cluster 13
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 12, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
18	Ottawa	12.0	Bus Station	Home Service	Restaurant	Yoga Studio	Electronics Store	Fried Chicken Joint	French Restaurant	Food Truck	Food & Drink Shop	Fast Food Restaurant

```python
# cluster 14
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 13, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
21	Ottawa	13.0	Cosmetics Shop	Park	Bar	Farmers Market	Furniture / Home Store	Frozen Yogurt Shop	Fried Chicken Joint	French Restaurant	Food Truck	Food & Drink Shop

```python
# cluster 15
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 14, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
9	Ottawa	14.0	Restaurant	Athletics & Sports	Lawyer	Yoga Studio	Electronics Store	Frozen Yogurt Shop	Fried Chicken Joint	French Restaurant	Food Truck	Food & Drink Shop

```python
# cluster 16
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 15, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
29	Ottawa	15.0	Hockey Arena	Pub	College Gym	Restaurant	Yoga Studio	Dive Bar	French Restaurant	Food Truck	Food & Drink Shop	Fast Food Restaurant

```python
# cluster 17
ottawa_merged.loc[ottawa_merged['Cluster Labels'] == 16, ottawa_merged.columns[[1] + list(range(5, ottawa_merged.shape[1]))]]
```

	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
20	Ottawa	16.0	Fast Food Restaurant	Restaurant	Park	Scenic Lookout	Yoga Studio	Dive Bar	French Restaurant	Food Truck	Food & Drink Shop	Farmers Market

Question 2 – results

Objective: A restaurant owner is looking to open a new Italian restaurant in Toronto, the objective is to recommend the best area in which a new restaurant could be located. In this section I will detail the results of my analysis – for the code please see my jupyter notebook.

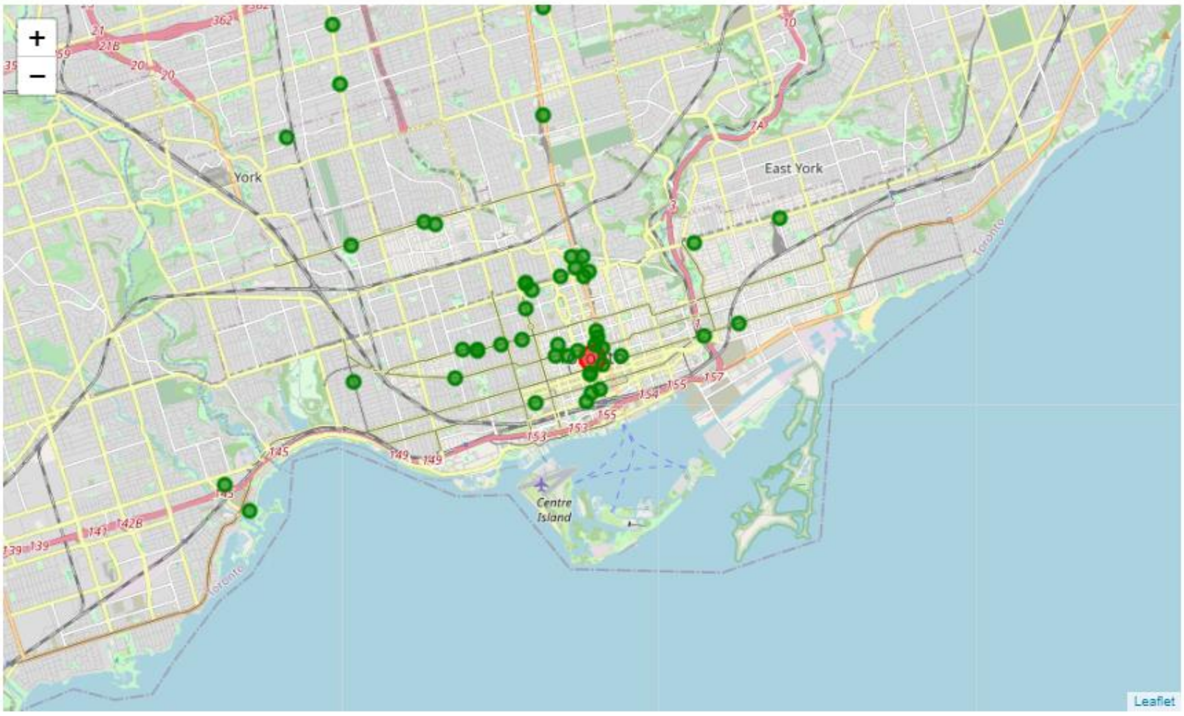
1. To prevent duplication the data used for this question is the same underlying Toronto dataset discussed in question 1. Therefore, from a results point of view this was already covered – please see question 1 results for the Toronto dataset above – points 1,2, and 3.
2. Search query for Italian restaurants and resulting dataframe.

```
dataframe_filtered
```

	name	categories	address	lat	lng	labeledLatLngs	distance	postalCode	cc	city	state	country	formattedAddress
0	Fabbrica Rustic Italian	Italian Restaurant	66 Wellington St W	43.647161	-79.381691	[{'label': 'display', 'lat': 43.647161, 'lng':...	726	M5K 1E7	CA	Toronto	ON	Canada	[66 Wellington St W, Toronto ON M5K 1E7, Canada]
1	Scaddabush Italian Kitchen & Bar	Italian Restaurant	382 Yonge Street, Unit #7	43.658920	-79.382891	[{'label': 'display', 'lat': 43.65892029202872...	611	M5B 1S8	CA	Toronto	ON	Canada	[382 Yonge Street, Unit #7 (Gerrard), Toronto ...
2	Mustachio Italian Eatery	Italian Restaurant	595 Bay St	43.656160	-79.383190	[{'label': 'display', 'lat': 43.65616, 'lng': ...	304	M5G 2C2	CA	Toronto	ON	Canada	[595 Bay St (Dundas St), Toronto ON M5G 2C2, C...
3	Punto Gelato, Simply Italian	Ice Cream Shop	146 Cumberland St	43.669955	-79.392603	[{'label': 'display', 'lat': 43.66995452843031...	1962	M5R 1A8	CA	Toronto	ON	Canada	[146 Cumberland St (btwn Avenue Rd & Bay St), ...
[illegible]	Elm Street	Italian	15 Elm	[illegible]	[illegible]	[{'label': 'display', 'lat':	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[15 Elm Street,

3. Visualisation – spatial location of all pre-existing Italian restaurants in Toronto.

```
# display map
venues_map
```



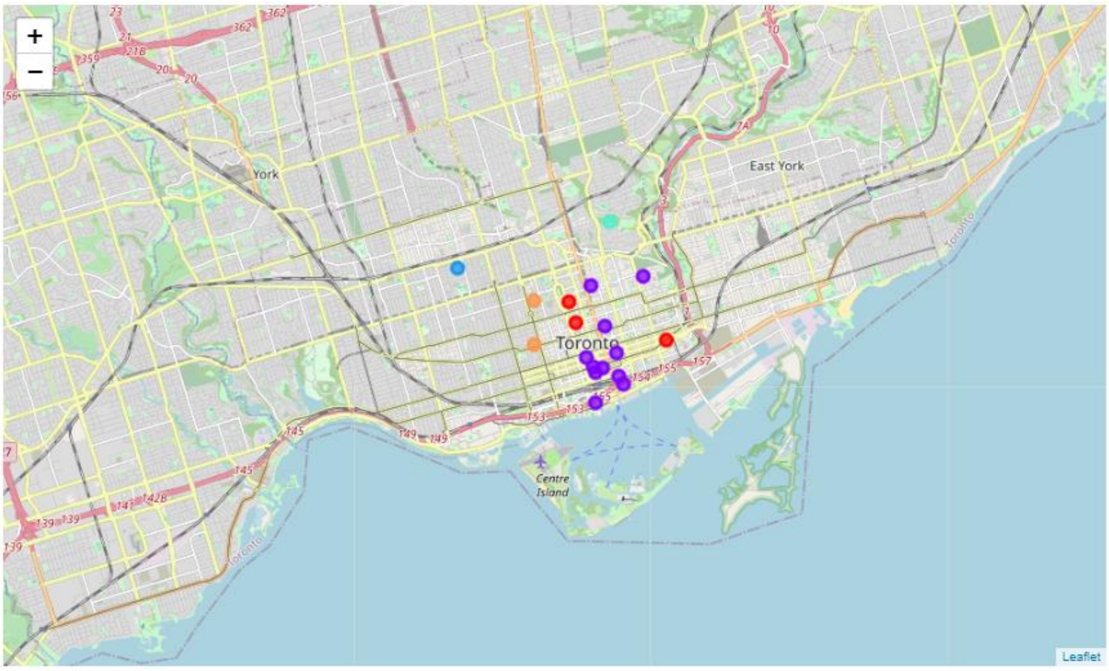
d. Discussion

Discussion section where you discuss any observations you noted and any recommendations you can make based on the results.

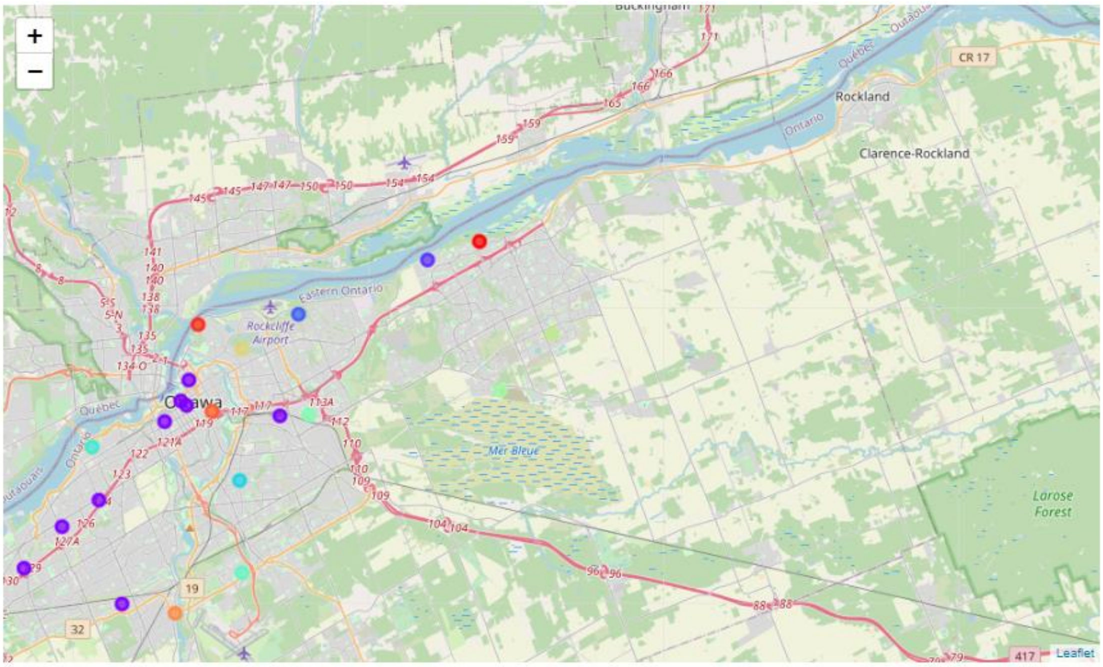
Question 1 – discussion:

Objective: to compare the neighbourhoods of Downtown Toronto and Ottawa and determine how similar or dissimilar they are.

Downtown Toronto:



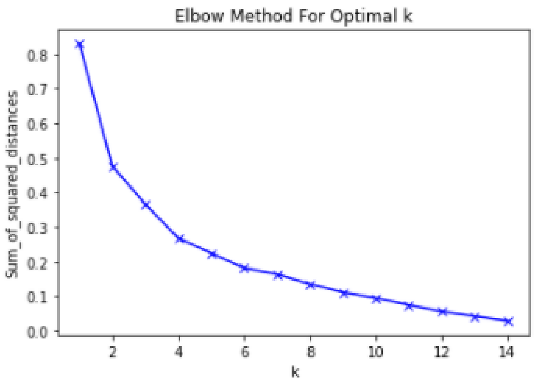
Ottawa:



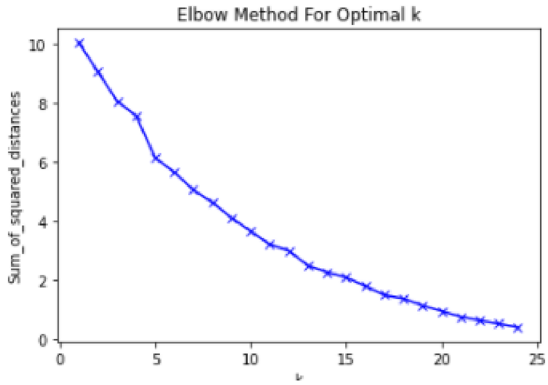
Observations:

1. The concentration of clusters is drastically different between the two locations chosen. This is a result of the Toronto dataset being isolated to an individual borough “Downtown Toronto”. In ideal world it would be appropriate to compare the two locations at a city level however the comparison performed is due to limitations in the dataset.
2. The value for of k in the k-means algorithm is also very different between the two locations. Showing that the locations are quite dissimilar – Toronto k = 6, Ottawa k = 17.

Downtown Toronto K



Ottawa K



3. Toronto dataset:
    a. The dominant cluster is 2. The most common venue in this cluster is a coffee shop which makes sense in a metropolis.
    b. The performance of the k-means clustering model has separated cluster 1 and cluster 2. However, on closer analysis of the data the most common venue in both clusters is a coffee shop, so they should in fact be a single cluster.
    c. The remaining clusters have performed well, separating out venues such as parks, airport, grocery stores and cafes.
4. Ottawa dataset:
    a. The dominant cluster is again 2 in this dataset. However, the performance of the model is poor in comparison to the Toronto dataset. Cluster 2 contains a mixture of venue categories ranging from coffee shops, gyms, pools and restaurants.
    b. The remaining clusters have performed substantially better, successfully categorising arenas, bus stations, construction & landscaping, and theme parks.

Recommendations:

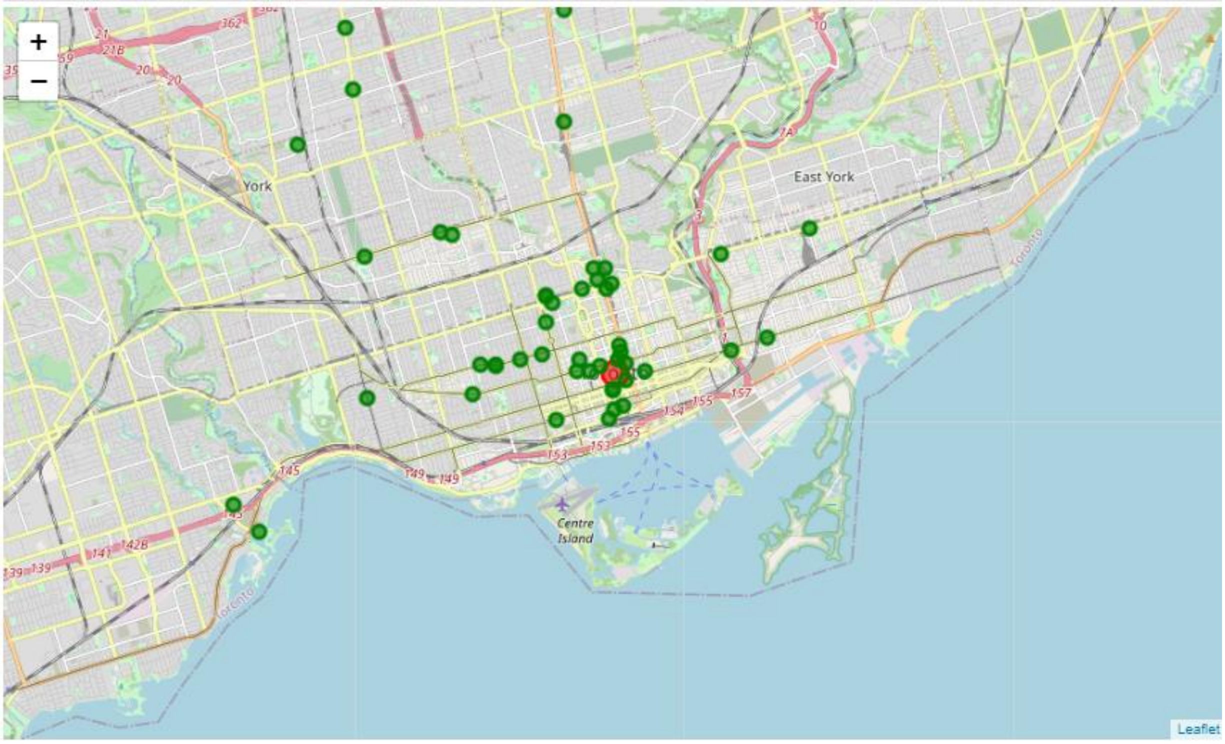
1. The high value of k in the Ottawa dataset promotes overfitting so the k-means method of clustering is not the greatest for predictive analysis of this dataset. However, the low value and clear elbow of k in the Toronto dataset does demonstrate a relatively robust model.
2. The above comment is justified by the success of the k-means algorithm in the Toronto dataset and its categorization of coffee shops.
3. I recommend using k-means algorithm when analysing the Toronto dataset, and a different clustering algorithm on the Ottawa dataset.

Question 2 discussion:

Objective: A restaurant owner is looking to open a new Italian restaurant in Toronto, the objective is to recommend the best area in which a new restaurant could be located.

The spatial distribution is highly important from a competition point of view as an area highly saturated in Italian cuisine will prove detrimental to their business. Therefore, the owner will be looking for an area that has none/few Italian restaurants at present.

Italian restaurants in Toronto:



Observations:

1. The distribution of Italian restaurants in Toronto is concentrated in the city centre, with venues reducing in frequency away from the centre of the city
2. The specific locations of Italian restaurants in Toronto does seem to be isolated to a few specific streets – demonstrated by the linear patterns.
3. The sporadic distribution of Italian restaurants on the outskirts of the city does indicate that other owners have taken a risk of building away from the city centre

Recommendations:

1. The location in which there is the most Italian restaurant competition is clearly the city centre. As the restaurant owner wants to minimise his competition building the new restaurant in this location would not be recommended.
2. Building on the outskirts of the city is a more risky endeavour are you could become isolated, and have minimal traffic into the restaurant.
3. I recommended choosing a location outside of the city centre and perform some additional analysis such as:

    a. Population movement – are there areas of high intensity on the outskirts of the city? Where are the choke points?
    b. Local population concentration – if the owner wants to only attract local customers.
4. Combing the additional recommended analysis will give the owner a more complete picture of the optimum location to build his restaurant.

e. CONCLUSION

Question 1 conclusion:

The neighbourhoods of Downtown Toronto and Ottawa a very dissimilar shown by the k-means algorithm detailed in this report.

Question 2 conclusion:

The best location for the owner of the Italian restaurant with his given criteria of little competition is on the outskirts of the city centre.

Please note the recommendations in the discussion section of this report for both questions.