The Battle of Neighborhoods - Restaurant's Location in London Ontario

Yicheng Huang

Dec 10, 2020

1. Introduction

In the previous course, I have learned how to use location data to explore the skills and tools of geographic location. In this project, I will innovate according to my needs and implement it based on the knowledge I have learned. Foursquare location data can explore or compare the neighborhood or city you choose or ask questions that can be solved using Foursquare location data. So Foursquare API will be an important part of this project.

In the previous course, we have done some research on Toronto and New York City. In this project, we will refer to the research methods in the previous examples and make a certain amount of expansion and innovation.

1.1. Business problem

I chose London Ontario as my research area in this project. London is a city in southwestern Ontario, Canada, along the Quebec City-Windsor corridor. According to the 2016 Canadian Census, the population of the city was 383,822. London is located at the confluence of the Thames, about 200 kilometers from Toronto and Detroit. The most important thing is that I study and live here, and the area where I study life will have a sense of substitution and accomplishment. The City of London is a competitive place, especially if you want to open a restaurant, then I want to help potential stakeholders to better understand the town and the market with useful insights. In this project, I hope to help them solve the following problems:

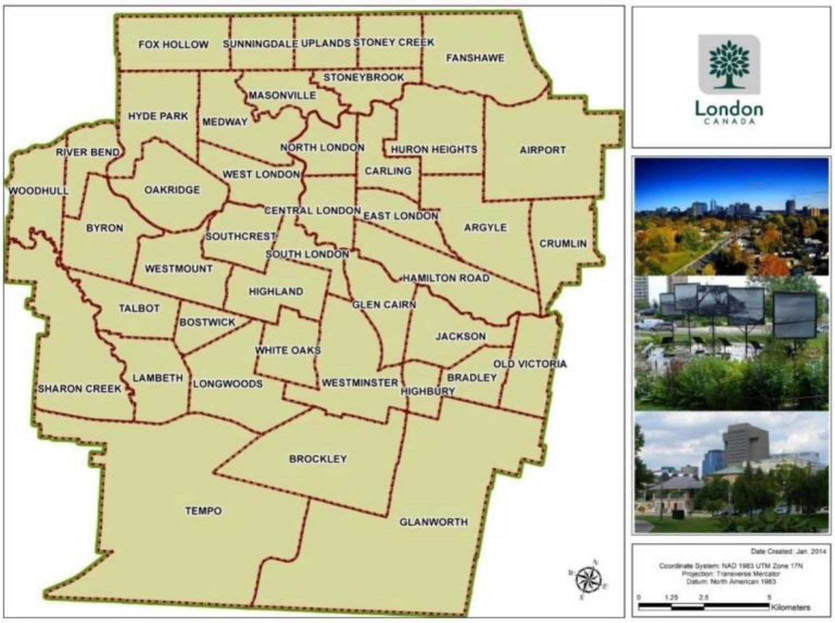
Where is the best place to open a restaurant in London, Ontario? And which type of restaurant is more popular?

1.2. Target audience

- Entrepreneurs who want to open a new restaurant in London.
- Hope to use python, Jupyter notebooks and some machine learning techniques to analyze business analysts or data scientists in the surrounding area of London.
- Some people are curious about the data they want to have an idea, how good it is to open a restaurant, and what are the pros and cons of this business.

2. Data Section

First of all, we need some geographic information about the London area, such as towns\regions, population, latitude\longitude, etc... Therefore, I think Wikipedia is the first place to check, as we learned in the course before I first looked up the postcode information of the London area on the wiki (we used the postcode information wiki page of the Toronto area in the course)
https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_N
Unfortunately, there is no borough and latitude and longitude data on the London page.Then I looked up the official map of City of London.
https://london.maps.arcgis.com/apps/webappviewer/index.html?id=0187f8a72f204edcbc95d595f31b5117



However, things are still not going well. Although I can see the map of London district on the website, I cannot download the available data on this website.
Finally, I found the available data on GeoNames.
http://www.geonames.org/postalcode-search.html?q=london&country=CA&adminCode1=ON
You can get the district represented by each postcode and their latitude and longitude data on this website. Because the latitude and longitude data is contained in a long list of descriptions, it is difficult to use beautifulsoup to extract the table, so I used MS Excel to integrate the website data and output it into a csv file.
I uploaded this CSV to IBM Cloud using the input code that comes with IBM Watson Studio, and can use the csv file in Jupyter notebooks with the following code.

```python
import types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
client_cdc84f548fe64aa191a1f7e326964da4 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='1K7D15K1feQ4tZRMQYuEdRzzYd3wH4pXm6ygto4wHpQz',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3-api.us-geo.objectstorage.service.networklayer.com')

body = client_cdc84f548fe64aa191a1f7e326964da4.get_object(Bucket='capstoneprojectnotebook-donotdelete-pr-z8zqh60vs0n8yx',Key='Londondistrict.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

df_data_3 = pd.read_csv(body)
df_data_3.head()
```

	Place	Code	Country	Admin1	Admin2	Admin3	latitude	Longitude
0	1	London (West Huron Heights / Carling)	N5Y	Canada	Ontario	London	43.012	-81.231
1	2	London (Glen Cairn)	N5Z	Canada	Ontario	London	42.966	-81.205
2	3	London (East Tempo)	N6L	Canada	Ontario	London	42.872	-81.247
3	4	London East (SW Argyle / Hamilton Road)	N5W	Canada	Ontario	London	42.986	-81.182
4	5	London West (Central Hyde Park / Oakridge)	N6H	Canada	Ontario	London	42.991	-81.340

Then I used a combination of geocoder.Nominatim and foursquare API to import the data of restaurants and their geographic information.

```python
address = 'London , Ontario'

geolocator = Nominatim(user_agent="LNON_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of London Ontario is {}, {}.'.format(latitude, longitude))
```

```python
# Set up Foursquare
CLIENT_ID = '[redacted]' # your Foursquare ID
CLIENT_SECRET = '[redacted]' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version
LIMIT = 100
print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```python
search_query = 'Restaurant'
radius = 50000
print(search_query + ' .... OK!')
```

```python
results = requests.get(url).json()
```

Next, we need to prepare a dataframe for K-means clustering. First, we need to combine the district data with the foursquare API data so that each restaurant can be classified into the neighborhood it belongs to. The code and result are following:

```python
def getNearbyVenues(names, latitudes, longitudes, radius=1000):

    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={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # 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)
```

```python
london_venues = getNearbyVenues(names=df_data_3['Code'],
                                   latitudes=df_data_3['latitude'],
                                   longitudes=df_data_3['Longitude']
                                  )
```

```python
london_venues.to_csv('london_venues.csv')
london_venues = pd.read_csv('london_venues.csv')
london_venues
```

	Unnamed: 0	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	0	London (West Huron Heights / Carling)	43.012	-81.231	Merla-Mae Ice Cream	43.010014	-81.241613	Ice Cream Shop
1	1	London (West Huron Heights / Carling)	43.012	-81.231	Jumbo Video	43.011495	-81.241666	Video Store
2	2	London (West Huron Heights / Carling)	43.012	-81.231	Metro	43.007334	-81.239668	Supermarket
3	3	London (West Huron Heights / Carling)	43.012	-81.231	Pho Lee	43.006657	-81.239302	Thai Restaurant
4	4	London (West Huron Heights / Carling)	43.012	-81.231	LCBO	43.003487	-81.227467	Liquor Store
5	5	London (West Huron Heights / Carling)	43.012	-81.231	Cora's	43.006783	-81.239025	Breakfast Spot
6	6	London (West Huron Heights / Carling)	43.012	-81.231	The Beer Store	43.010916	-81.241374	Beer Store
7	7	London (West Huron Heights / Carling)	43.012	-81.231	RBC Royal Bank	43.011931	-81.241519	Bank
8	8	London (West Huron Heights / Carling)	43.012	-81.231	Subway	43.007322	-81.238760	Sandwich Place
9	9	London (West Huron Heights / Carling)	43.012	-81.231	Shoppers Drug Mart	43.012133	-81.243292	Pharmacy
10	10	London (West Huron Heights / Carling)	43.012	-81.231	Petro-Canada	43.011629	-81.242390	Gas Station
11	11	London (West Huron Heights / Carling)	43.012	-81.231	Kelseys Original Roadhouse	43.003319	-81.228502	Restaurant

Then I use 'one hot coding' and 'group' created an dataset that can be used for K-mean clustering.

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

# add neighborhood column back to dataframe
london_onehot['Neighborhood'] = london_venues['Neighborhood']

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

```python
london_grouped = london_onehot.groupby('Neighborhood').mean().reset_index()
```

	Neighborhood	African Restaurant	American Restaurant	Asian Restaurant	Automotive Shop	Baby Store	Bakery	Bank	Bar
0	London (East Tempo)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	London (Fanshawe / Stoneybrook / Stoney Creek ...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	London (Glen Cairn)	0.000000	0.066667	0.000000	0.000000	0.000000	0.066667	0.066667	0.000000
3	London (Riverbend / Woodhull / North Sharon Cr...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.076923	0.000000
4	London (South White Oaks / Central Westminster...	0.000000	0.250000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
5	London (Southcrest / East Westmount /	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.142857	0.000000

	Neighborhood	Beer Store	Bookstore	Breakfast Spot	Brewery	Burrito Place	Business Service	Café	Chinese Restaurant	Church	Clothing Store
0	London (East Tempo)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	London (Fanshawe / Stoneybrook / Stoney Creek ...	0.000000	0.000000	0.250000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	London (Glen Cairn)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	London (Riverbend / Woodhull / North Sharon Cr...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.153846	0.000000	0.000000
4	London (South White Oaks / Central Westminster...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
5	London (Southcrest / East Westmount /	0.000000	0.000000	0.000000	0.000000	0.000000	0.142857	0.000000	0.000000	0.000000	0.000000

3. Methodology

3.1. Business understanding

The purpose of the project is to find the best community in London to open a new restaurant.

3.2. Analytical method

The total number of communities in London is 14, so we need to find a way to cluster them based on similarity (that is, the number and type of restaurants). Briefly, after some steps of data cleaning and data exploration, I will use the K-Means algorithm to extract clusters, generate graphs, and demonstrate the final results.

3.3. Data exploration

To explore the data, I will use "Folium" a python library that can create interactive maps using coordinate data.
In order to complete this map, we must first process the data we obtained so that these data can be applied to "Folium".
First, we need to convert JSON into a dataframe, and then extract the information we need from the initial dataframe.

```
# assign relevant part of JSON to venues
venues = results['response']['venues']

# tranform venues into a dataframe
dataframe = json_normalize(venues)
```

```python
# keep only columns that include venue name, and anything that is associated with location
filtered_columns = ['name', 'categories'] + [col for col in dataframe.columns if col.startswith('location.')] + ['id']
dataframe_filtered = dataframe.loc[:, filtered_columns]

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

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

# filter the category for each row
dataframe_filtered['categories'] = dataframe_filtered.apply(get_category_type, axis=1)

# clean column names by keeping only last term
dataframe_filtered.columns = [column.split('.')[-1] for column in dataframe_filtered.columns]

dataframe_filtered.head()
```

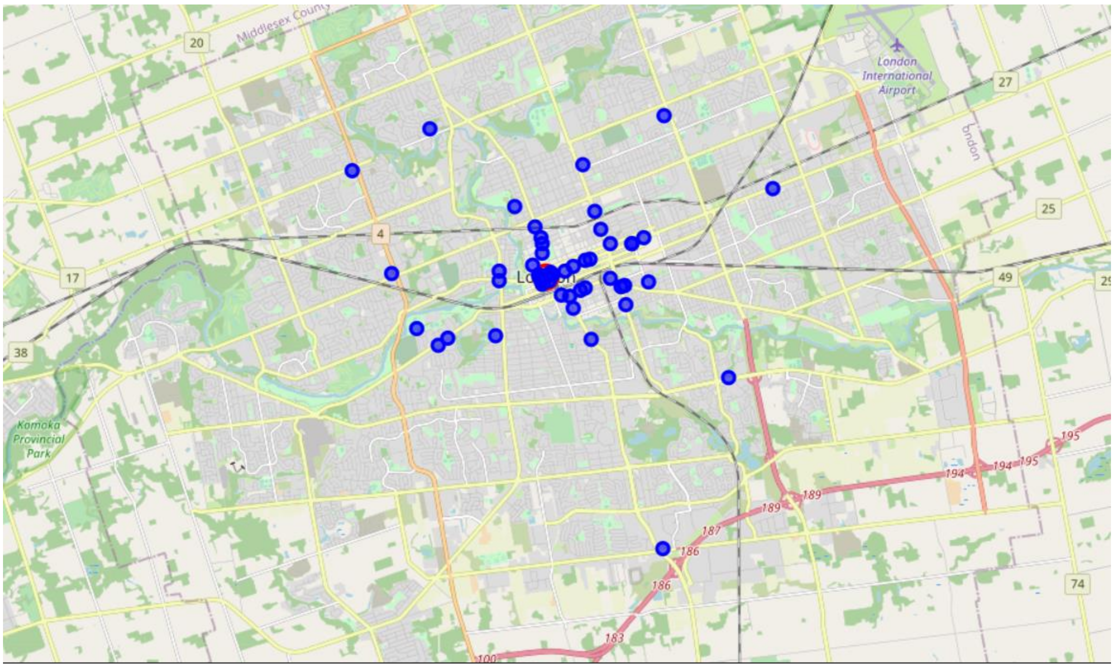
	name	categories	address	crossStreet	lat	lng	labeledLatLngs	distance	postalCode	cc	city	state	country	formattedAddress	neighborhood	id
0	Thaifoon Restaurant	Thai Restaurant	120 Dundas St.	near Talbot St.	42.983311	-81.251357	[{'label': 'display', 'lat': 42.98331116257564...	148	N6A 1G3	CA	London	ON	Canada	[120 Dundas St. (near Talbot St.), London ON N...	NaN	4b97329ff964a520cbfb34e3
1	Tahinis Restaurant	Middle Eastern Restaurant	11-551 Richmond St	at Albert St.	42.988600	-81.250732	[{'label': 'display', 'lat': 42.9885998, 'lng'...	555	N6A 3E9	CA	London	ON	Canada	[11-551 Richmond St (at Albert St.), London ON...	NaN	4b61ce40f964a520fa232ae3
2	Melody Restaurant	Diner	NaN	NaN	42.983746	-81.249484	[{'label': 'display', 'lat': 42.983746, 'lng':...	12	NaN	CA	London	ON	Canada	[London ON, Canada]	NaN	4e74e1fb8998ed82a43fcaf0
3	Symposium Cafe Restaurant & Lounge	Restaurant	620 Richmond St.	at Central Ave.	42.990655	-81.250828	[{'label': 'display', 'lat': 42.990655149613, ...	783	N6J 5A9	CA	London	ON	Canada	[620 Richmond St. (at Central Ave.), London ON...	NaN	4b5e63e5f964a520ba8c29e3
4	Westside Family Restaurant	Breakfast Spot	107 Mount Pleasant Ave	Wharncliffe Rd. N.	42.984838	-81.263275	[{'label': 'display', 'lat': 42.98483774629121...	1120	N6H 1E1	CA	London	ON	Canada	[107 Mount Pleasant Ave (Wharncliffe Rd. N.), ...	NaN	4bf532dd706e20a1e222aa98

Now we can use folium to visualize the location of the recorded restaurant.

```python
venues_map = folium.Map(location=[latitude, longitude], zoom_start=13)


folium.CircleMarker(
    [latitude, longitude],
    radius=10,
    color='red',
    popup='London Ontario',
    fill = True,
    fill_color = 'red',
    fill_opacity = 0.6
).add_to(venues_map)


for lat, lng, label in zip(dataframe_filtered.lat, dataframe_filtered.lng, dataframe_filtered.categories):
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        color='blue',
        popup=label,
        fill = True,
        fill_color='blue',
        fill_opacity=0.6
    ).add_to(venues_map)
```



Before to continue, it could be a good idea to check what kind of venue are popular in London.

```
londonCt = london_venues.groupby('Venue Category').size().reset_index(name='counts')
londonCt.sort_values(by=['counts'], ascending=False).head(10)
```

]:

	Venue Category	counts
18	Coffee Shop	12
65	Sandwich Place	8
31	Grocery Store	7
7	Bar	6
59	Pizza Place	6
58	Pharmacy	6
63	Restaurant	6
56	Park	6
20	Convenience Store	6
37	Hotel	6

From this, it can be clearly seen that coffee shops are the most popular among the recorded locations that can provide food.

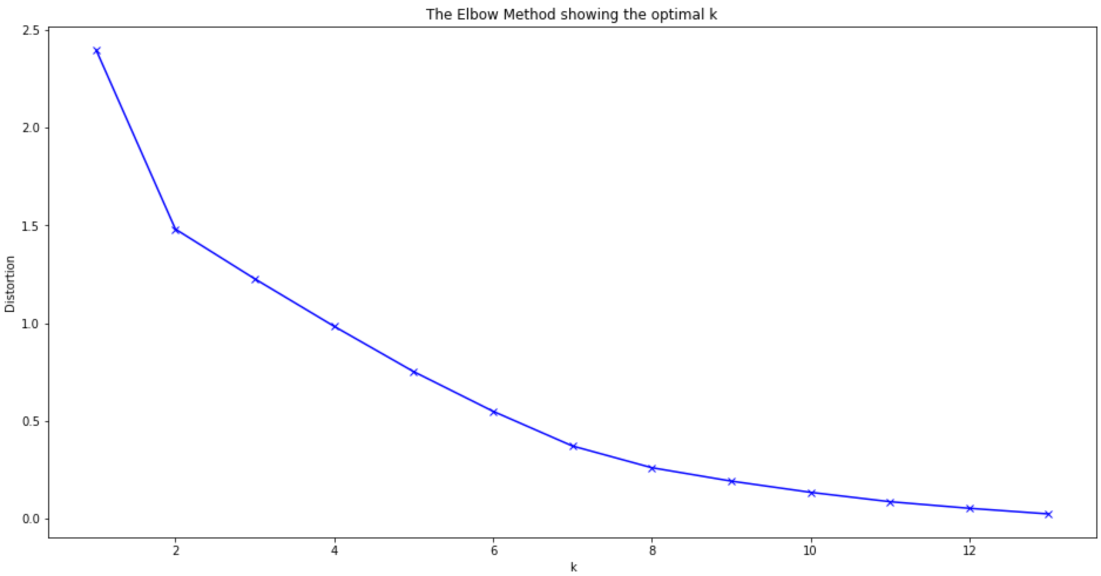
3.4. Clustering

To analyze which neighbourhood in London is suitable for opening a new restaurant, I will use K-means clustering: an unsupervised learning, used when you have unlabeled data (that is, data with no defined categories or groups). The goal of this algorithm is to find groups in the data, and the number of groups is represented by the variable K. The algorithm iteratively assigns each data

point to one of K groups according to the provided function. Data points are clustered based on feature similarity.
Therefore, the first step is to use the well-known analysis method-the ‘elbow method” to determine the best "K".

```
%matplotlib inline
import matplotlib.pyplot as plt

london_grouped_clustering = london_grouped.drop('Neighborhood', 1)
distortions = []
K = range(1,14)
for k in K:
    kmeanModel = KMeans(n_clusters=k)
    kmeanModel.fit(london_grouped_clustering)
    distortions.append(kmeanModel.inertia_)
plt.figure(figsize=(16,8))
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
```



From the plot above, we can assume the suitable K = 8. Then we run the kmean- clustering by using the ‘K = 8’.

```
# run k-means clustering
kmeans = KMeans(n_clusters=8,random_state=0).fit(london_grouped_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:14]
```

```
]: array([1, 5, 0, 0, 7, 6, 0, 0, 4, 0, 3, 0, 0, 2], dtype=int32)
```

Then we get the ‘most common venues’ dataframe by using the following code.

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

    return row_categories_sorted.index.values[0:num_top_venues]

num_top_venues = 8

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

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

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

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

neighborhoods_venues_sorted.head()
```

And merge to obtain the final dataset and drop the NaN value in the dataframe:

```python
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

london_merged=df_data_3

london_merged = london_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Code')
london_merged = london_merged.rename(columns={'Code': 'Neighbourhood', 'Country': 'Postal Code'})
london_merged["Cluster Labels"] = london_merged["Cluster Labels"].fillna(0.0).astype(int)
london_merged.dropna(subset = ["1st Most Common Venue"], inplace=True)
london_merged.head() # check the last columns!
```

	Place	Neighbourhood	Postal Code	Admin1	Admin2	Admin3	latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
0	1	London (West Huron Heights / Carling)	N5Y	Canada	Ontario	London	43.012	-81.231	0	Video Store	Grocery Store	Pizza Place	Beer Store	Bookstore	Restaurant	Chinese Restaurant	Sandwich Place
1	2	London (Glen Cairn)	N5Z	Canada	Ontario	London	42.966	-81.205	0	Massage Studio	Grocery Store	Discount Store	Convenience Store	Salon / Barbershop	Coffee Shop	Skating Rink	Fast Food Restaurant
2	3	London (East Tempo)	N6L	Canada	Ontario	London	42.872	-81.247	1	Construction & Landscaping	Yoga Studio	Department Store	Discount Store	Event Space	Farmers Market	Fast Food Restaurant	French Restaurant
3	4	London East (SW Argyle / Hamilton Road)	N5W	Canada	Ontario	London	42.986	-81.182	3	Portuguese Restaurant	Construction & Landscaping	Park	Business Service	Music Store	Yoga Studio	Gastropub	Farmers Market
4	5	London West (Central Hyde Park / Oakridge)	N6H	Canada	Ontario	London	42.991	-81.340	2	Vineyard	Event Space	Market	Hardware Store	Yoga Studio	Golf Course	Discount Store	Farmers Market

4. Result and Discussion

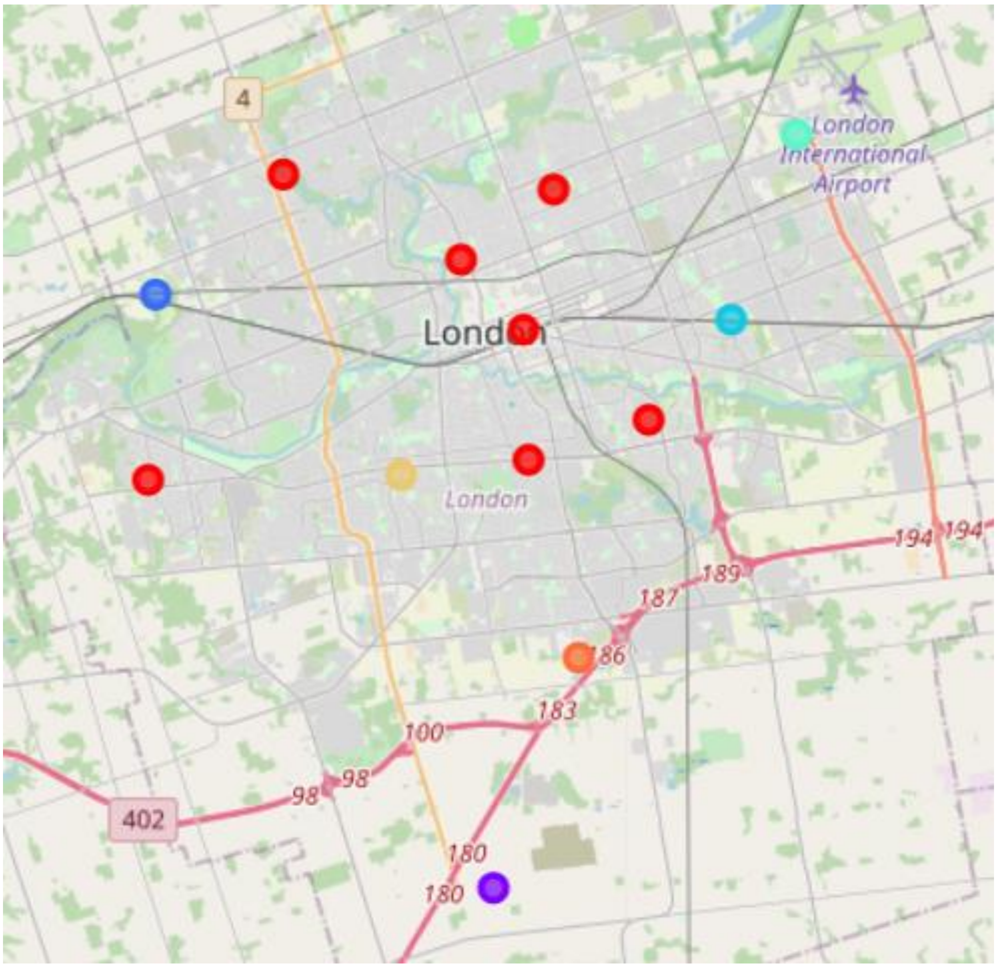
Before to start to analyze all the clusters, let us visualize and take a look on a folium map:

```python
kclusters = 8
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

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

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(london_merged['latitude'], london_merged['Longitude'], london_merged['Neighbourhood'], london_merged['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=0.7).add_to(map_clusters)

map_clusters
```



As we can see, each cluster belong to a color with different characteristics. You can read the complete list below:

Cluster 1 (Label = 0)

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

	Neighbourhood	Admin3	latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
0	London (West Huron Heights / Carling)	London	43.012	-81.231	0	Video Store	Grocery Store	Pizza Place	Beer Store	Bookstore	Restaurant	Chinese Restaurant	Sandwich Place
1	London (Glen Cairn)	London	42.966	-81.205	0	Massage Studio	Grocery Store	Discount Store	Convenience Store	Salon / Barbershop	Coffee Shop	Skating Rink	Fast Food Restaurant
9	London South (East Highland / North White Oaks...	London	42.958	-81.238	0	Coffee Shop	American Restaurant	Middle Eastern Restaurant	Chinese Restaurant	Sandwich Place	Restaurant	Pizza Place	Pharmacy
11	London (Sunningdale / West Masonville / Medway...	London	43.015	-81.305	0	Pharmacy	Pizza Place	Vietnamese Restaurant	History Museum	Baby Store	Toy / Game Store	Salon / Barbershop	Coffee Shop
12	London (Riverbend / Woodhull / North Sharon Cr...	London	42.954	-81.342	0	Convenience Store	Chinese Restaurant	Restaurant	Liquor Store	Pharmacy	Coffee Shop	Park	Supermarket
15	London Central	London	42.984	-81.239	0	Bookstore	Hotel	Vegetarian / Vegan Restaurant	Café	Indian Restaurant	Pub	Italian Restaurant	Gym / Fitness Center
16	London North (UWO)	London	42.998	-81.256	0	Bar	Sandwich Place	Coffee Shop	Yoga Studio	Restaurant	Breakfast Spot	Café	Irish Pub

Cluster 2 (Label = 1)

	Neighbourhood	Admin3	latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
2	London (East Tempo)	London	42.872	-81.247	1	Construction & Landscaping	Yoga Studio	Department Store	Discount Store	Event Space	Farmers Market	Fast Food Restaurant	French Restaurant

Cluster 3 (Label = 2)

	Neighbourhood	Admin3	latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
4	London West (Central Hyde Park / Oakridge)	London	42.991	-81.34	2	Vineyard	Event Space	Market	Hardware Store	Yoga Studio	Golf Course	Discount Store	Farmers Market

Cluster 4 (Label = 3)

	Neighbourhood	Admin3	latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
3	London East (SW Argyle / Hamilton Road)	London	42.986	-81.182	3	Portuguese Restaurant	Construction & Landscaping	Park	Business Service	Music Store	Yoga Studio	Gastropub	Farmers Market

Cluster 5 (Label = 4)

	Neighbourhood	Admin3	latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
7	London (YXU / North and East Argyle / East Hur...	London	43.023	-81.164	4	Golf Course	Coffee Shop	Soccer Stadium	Historic Site	Yoga Studio	Discount Store	Event Space	Farmers Market

Cluster 6 (Label = 5)

	Neighbourhood	Admin3	latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
8	London (Fanshawe / Stoneybrook / Stoney Creek ...	London	43.044	-81.239	5	Breakfast Spot	Trail	Park	Gas Station	Yoga Studio	Discount Store	Event Space	Farmers Market

Cluster 7 (Label = 6)

	Neighbourhood	Admin3	latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
5	London (Southcrest / East Westmount / West Hig...	London	42.955	-81.273	6	Grocery Store	Ice Cream Shop	Bank	Park	Sporting Goods Shop	Gas Station	Business Service	Gastropub

Cluster 8 (Label = 7)

	Neighbourhood	Admin3	latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
10	London (South White Oaks / Central Westminster...	London	42.918	-81.224	7	Intersection	American Restaurant	Hotel	Department Store	Golf Course	Discount Store	Event Space	Farmers Market

Therefore, we have the opportunity to have some discussion about the cluster. Let's see what we found:

- The most common places to eat in London are coffee shop and sandwich place.
- After removing all Columns containing NaN values, except for Cluster 1 (Label 0, the red dot on the map), all other clusters have only one district. So we focus on Cluster 1.
- Judging from the geographical representation of the cluster, most of the cluster 1 is near Western University (University of Western Ontario), so it is indeed a reasonable choice to open a coffee shop in these places.
- If our stakeholders think that there are too many coffee shops, they can also suggest that it is feasible to open fast-food restaurants or Chinese restaurants in these areas since these 2 are also very popular in this cluster.
- In addition, because only the Foursquare API is used, there are no suitable merge results in three districts (most common venues are all NaN), and there are some problems with the classification of restaurants, such as "restaurant" and "fast food restaurant". "And "sandwich place", categories like "restaurant" and "fast food restaurant" can be subdivided, so we need to find more detailed and precise data in the future.

In the future, I will try to use more different methods to establish a realistic and accurate data analysis program, such as: web crawling on various websites, new open data from the Public Administration Department (City of London), some powerful Python libraries etc. Folium and GeoPandas, Foursquare API, and other well-known APIs, etc...

5. Conclusion

Since the project only analyzed a small amount of data, we can get better results by adding neighborhood information. In any case, London is a livable city that offers many different types of new restaurant businesses. I think we have experienced identifying business problems, finding the required data, cleaning the data set, and executing machine learning algorithms using k-means clustering. process. And provide some useful tips to our stakeholders.

Reference

http://www.geonames.org/postalcode-search.html?q=london&country=CA&adminCode1=ON
https://london.maps.arcgis.com/apps/webappviewer/index.html?id=0187f8a72f204edcbc95d595f31b5117
https://london.ctvnews.ca/find-your-info-a-statistical-portrait-of-london-s-neighbourhoods-1.2033843