IBM Data Capstone Project – The Battle of Neighborhoods

Chengyu Wang

July 26, 2020

Introduction

Background

Toronto is Canada's largest city and a world leader in areas such as business, finance, technology, entertainment and culture. Its large population of immigrants from all over the globe has also made Toronto one of the most multicultural cities in the world.

The global COVID-19 pandemic has affected several countries in the world with Canada being one of them too. Toronto which is the 4th most populous city in North America is vastly affected too.

Toronto's chief medical officer of health says COVID-19 is present in every neighbourhood but some neighborhoods have higher number of infections.

Problem

The big question is which neighborhood should one choose in Toronto amidst this pandemic?

In this project, I will discuss different parameters to consider and use foursquare location data and clustering of venue information to determine a less impacted neighborhood in Toronto.

Target Audience

Anyone who is planning to move to the city of Toronto.

Students in Toronto who are looking for off-campus housing.

Anyone who is planning to move to another location in the city of Toronto.

Data

The data used in this project include:

1. City of Toronto COVID-19 Summary
   Sources: Ontario Ministry of Health, Integrated Public Health Information System and CORES
   Open Government License Toronto

   https://www.toronto.ca/home/covid-19/covid-19-latest-city-of-toronto-news/covid-19-status-of-cases-in-toronto/

   https://open.toronto.ca/dataset/covid-19-cases-in-toronto/

2. Venues in each neighborhood in the City of Toronto
   Source: Foursquare API

3. Toronto Neighborhoods details with respective CDN numbers
   Source: Wikipedia
   https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M

4. Toronto Neighborhoods details with respective Postal Codes
   Source: Wikipedia
   https://en.wikipedia.org/wiki/List_of_city-designated_neighbourhoods_in_Toronto

5. Geographical coordinates of the neighbourhoods
   Source: https://cocl.us/Geospatial_data

Data Cleaning

The data for the neighborhoods had many rows with neighborhoods that were not assigned. To counter this problem, I made the neighborhood the same as the borough if it was not assigned. The following code snippet shows how I achieved this.

```
df_postal.loc[df_postal['Neighborhood']=="Not assigned",'Neighborhood']=df
_postal.loc[df_postal['Neighborhood']=="Not assigned",'Borough']
df_postal.rename(columns={'Neighborhood':'Neighborhoods'}, inplace = True)
```

The datasets were merged into one dataframe based on the columns that they had in common. The final merged dataframe is shown below.

	Postal Code	Borough	Neighborhoods	Latitude	Longitude	CDN	Neighborhood	Case Count	Rate per 100,000 people
0	M3A	North York	Parkwoods	43.753259	-79.329656	42.0	Banbury-Don Mills	37	133.598122
1	M3A	North York	Parkwoods	43.753259	-79.329656	34.0	Bathurst Manor	126	793.800794
2	M3A	North York	Parkwoods	43.753259	-79.329656	52.0	Bayview Village	33	154.234436
3	M3A	North York	Parkwoods	43.753259	-79.329656	49.0	Bayview Woods-Steeles	116	881.861031
4	M3A	North York	Parkwoods	43.753259	-79.329656	39.0	Bedford Park-Nortown	81	348.597005

Methodology

Exploratory Data Analysis

The data from https://open.toronto.ca/dataset/covid-19-cases-in-toronto/

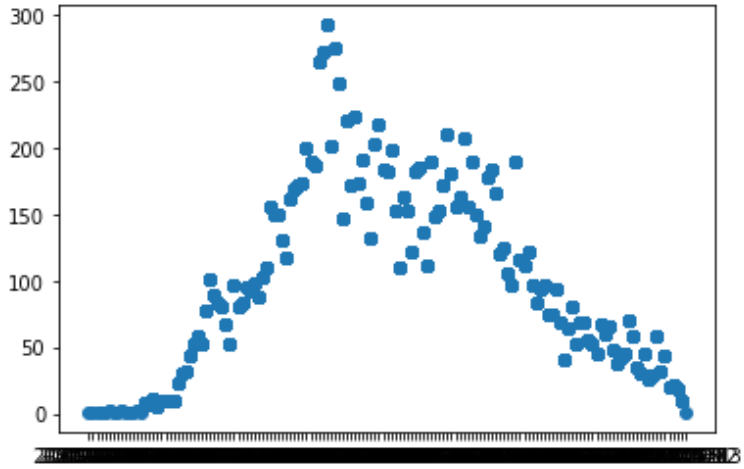
consisted of episode date and reported date. As only the data relating to the confirmed cases was extracted, I decided to work with the episode date.

The total number of cases per episode date was obtained using the following code:

```
total_cases = df_cases['Episode Date'].value_counts()
total_cases = pd.DataFrame(total_cases) total_cases.reset_index(level=0,
inplace=True)
total_cases.columns = ['Episode Date','Total Cases']
total_cases.sort_values('Episode Date',inplace=True)
```

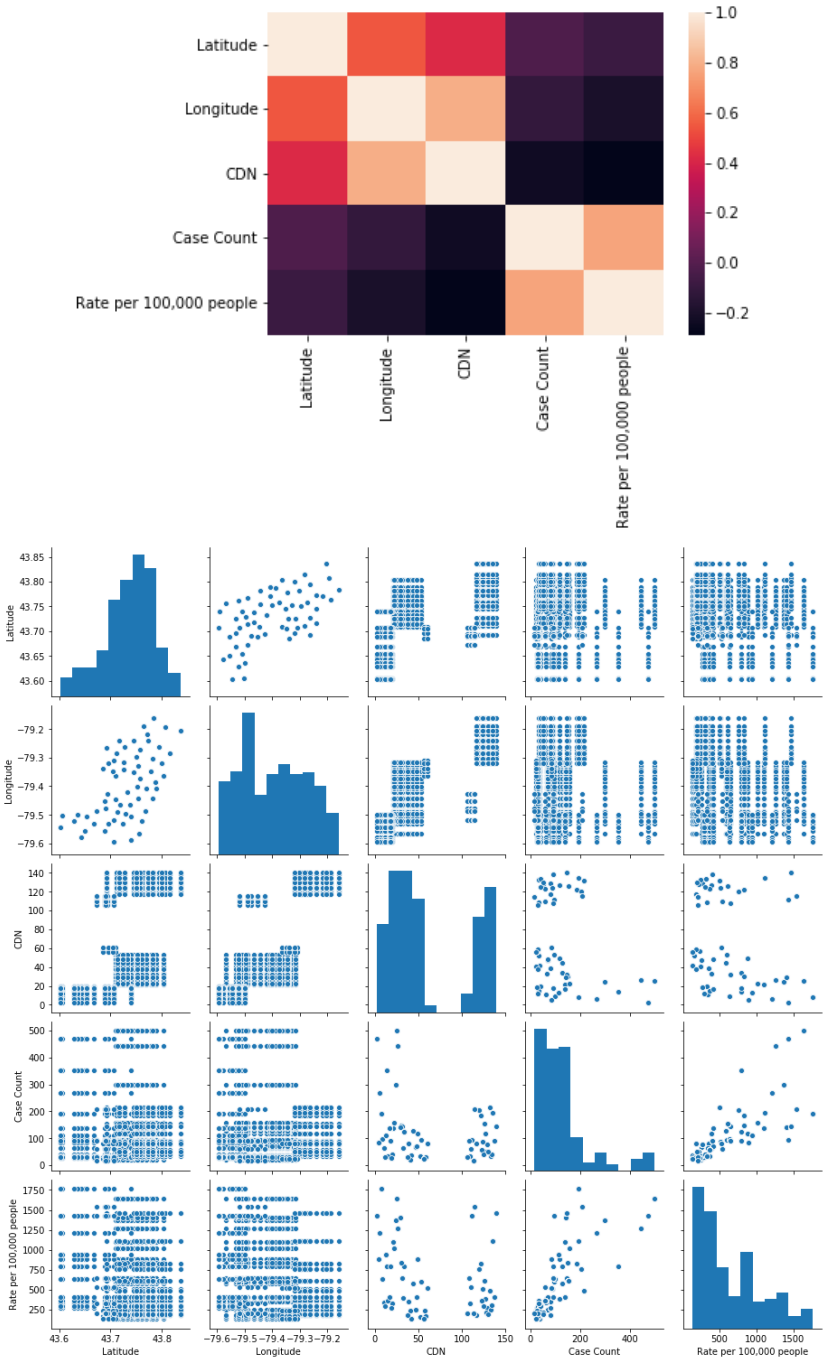
The column for the total cases was added to the original dataframe.

A scatter plot was obtained to analyze the relationship between total number of cases and the episode date:



From the scatter plot I could conclude that the relationship was nonlinear and the total number of cases in Toronto neighborhood had already reached its peak. However, this analysis is subject to change as it only comprises of data up to July 2020.

A heat map and a pairplot were made of all the factors to visually gauge average correlation of COVID-19 cases and factors in the dataset.



From the pairplot we can roughly tell that the neighborhoods (CDN numbers) were clustered based on the number of cases and rate per 100,000 people.

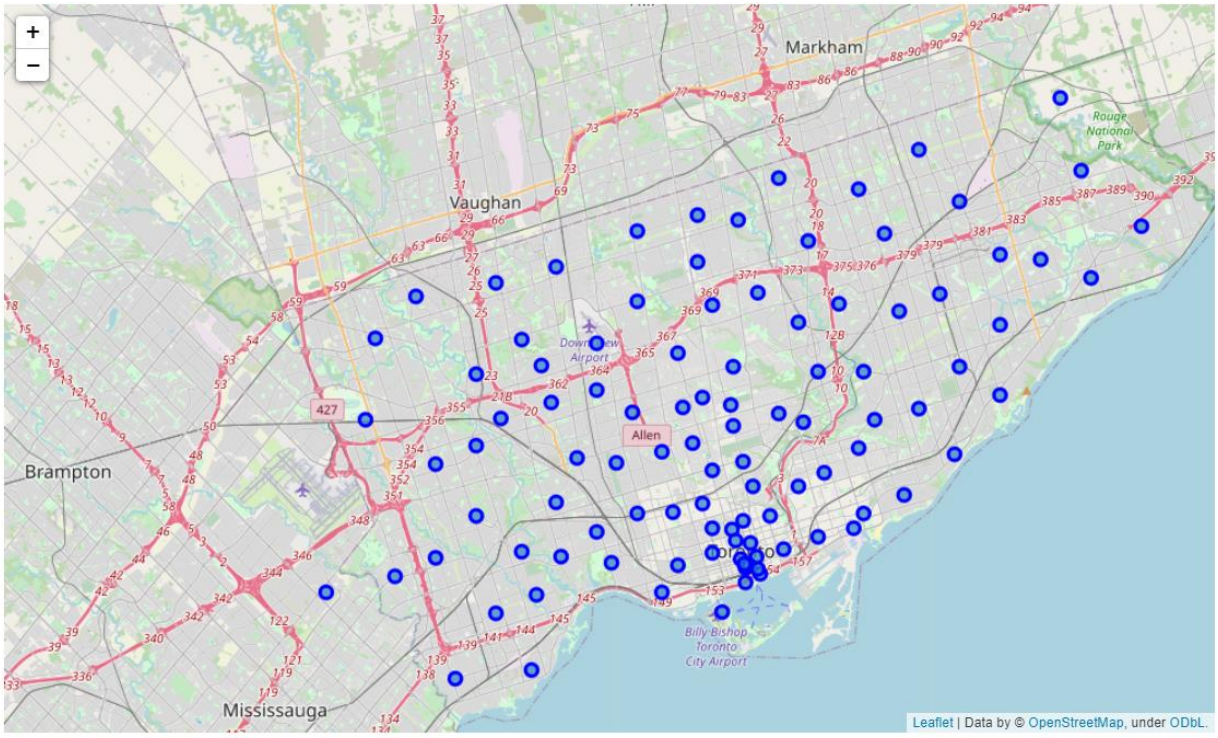
Geolocation

Google maps geocoder API was used to obtain the latitude and longitude values of the city of Toronto.

```
address = 'Toronto ON'
geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinates of Toronto are {},
{}.'.format(latitude, longitude))
```

Folium

Folium was used to visualize the data in an interactive leaflet map. A map of the city of Toronto with the neighborhoods superimposed on it was created.



Foursquare API

I utilized the Foursquare API to explore the boroughs and search for hospitals. I searched for hospitals around each borough in a 100 meter radius from their given latitude and longitude values.

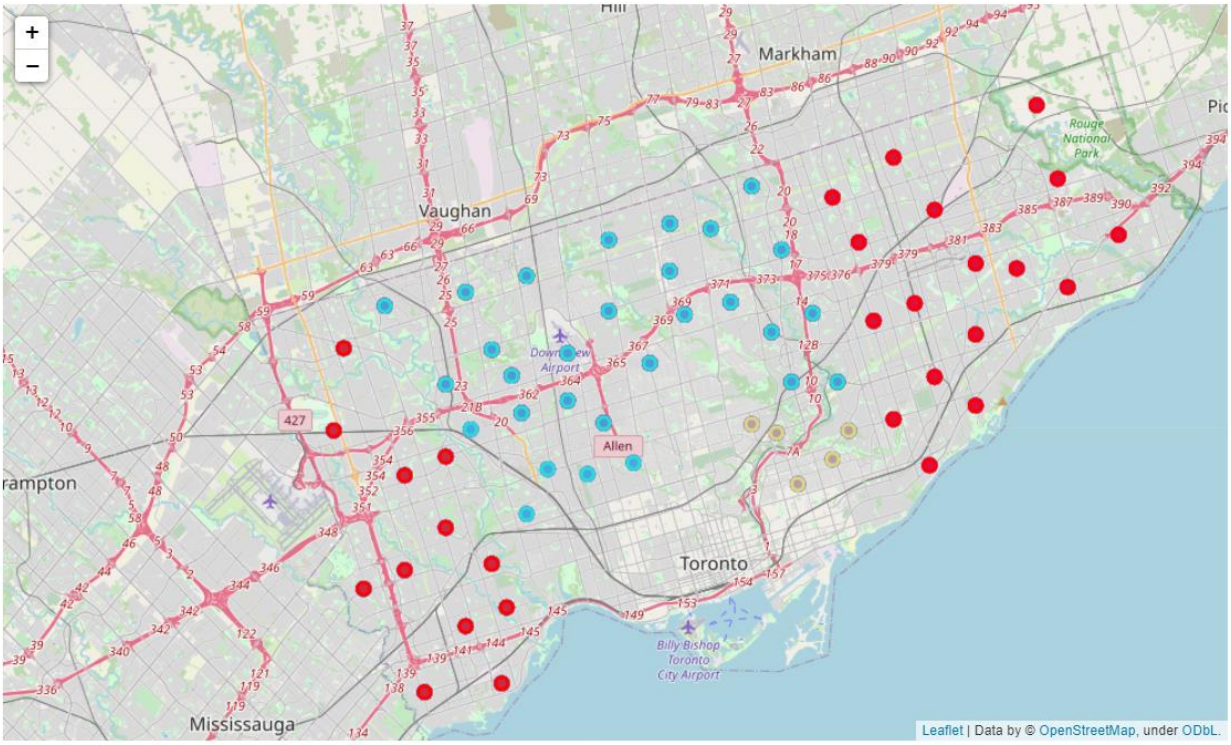
```
def getNearbyVenues(names, latitudes, longitudes, radius=100):

    categoryId = '4bf58dd8d48988d196941735' # category for hospitals
    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/search?&client_id={}&c
lient_secret={}&v={}&ll={},{}&radius={}&limit={}&categoryId={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT,
            categoryId)

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

        # 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']) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for i
tem in venue_list])

    return nearby_venues
```

K Means Clustering

As in the data exploratory section I observed that there was a likelihood that neighborhoods were clustered based on cases and rate per 100,000 people I decided to use the K Means Machine Learning Algorithm to train the data and get the clusters.
After running K Means a number of times I observed that the optimal value for K was 4.

```
df_features = df_merged[['Rate per 100,000 people','Case Count']]
# set number of clusters
kclusters = 4
# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(df_features)
# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
df_merged['Cluster Labels'] = kmeans.labels_
```

Once again Folium was used to generate a map of the neighborhoods in Toronto but this time the markers were color coded based on the clusters obtained from the K Means algorithm.



Results and Discussion

On the evaluation of the clusters I devised suitable labels for each cluster:

In Cluster labeled **1** the case count was **below 150** for all neighborhoods and the rate per 100,000 approximately lied between **100-420**. Therefore, its label name would be "***Least Infected***".

In Cluster labeled **3** the case count was **below 300** for all neighborhoods and the rate per 100,000 approximately lied between **450-700**. Therefore, its label name would be "***Considerably Infected***".

In Cluster labeled **0** the case count was **below 400** for all neighborhoods and the rate per 100,000 approximately lied between **700-1200**. Therefore, its label name would be "***Highly Infected***".

In Cluster labeled **2** the case count was **below 500** for all neighborhoods and the rate per 100,000 was **above 1200** for all neighborhoods. Therefore, its label name would be "***Extremely Infected***".

Conclusion

In this study after carefully analyzing COVID-19 in the city of Toronto from various datasets and using various visualizations and K Means Clustering I clustered neighborhoods based on case count and rate per 100,000.

Currently COVID-19 has a significant impact globally and therefore it is a major factor to take into account when deciding a neighborhood. Therefore, this model is useful to narrow down one's choice of neighborhood in the city of Toronto.

However, this model only clusters the neighborhoods based on the COVID-19 pandemic and other factors need to be considered when deciding a neighborhood for instance the price, distance from workplace/school, safety etc.

As I mentioned earlier the data is subject to change thus the results of this model are also subject to change.

Acknowledgements

I sincerely thank all the course instructors who took their time and effort into making this Applied Data Science Specialization Certificate.

Thanks to all the peer reviewers who graded my projects.