In [6]:
```
! pip install nbconvert
```

```
Requirement already satisfied: nbconvert in /home/jupyterlab/conda
/envs/python/lib/python3.6/site-packages (6.0.7)
Requirement already satisfied: nbformat>=4.4 in /home/jupyterlab/c
onda/envs/python/lib/python3.6/site-packages (from nbconvert) (5.0
.8)
Requirement already satisfied: jupyter-core in /home/jupyterlab/co
nda/envs/python/lib/python3.6/site-packages (from nbconvert) (4.7.
0)
Requirement already satisfied: entrypoints>=0.2.2 in /home/jupyter
lab/conda/envs/python/lib/python3.6/site-packages (from nbconvert)
(0.3)
Requirement already satisfied: jinja2>=2.4 in /home/jupyterlab/con
da/envs/python/lib/python3.6/site-packages (from nbconvert) (2.11.
2)
Requirement already satisfied: bleach in /home/jupyterlab/conda/en
vs/python/lib/python3.6/site-packages (from nbconvert) (1.5.0)
Requirement already satisfied: testpath in /home/jupyterlab/conda/
envs/python/lib/python3.6/site-packages (from nbconvert) (0.4.4)
Requirement already satisfied: pandocfilters>=1.4.1 in /home/jupyt
erlab/conda/envs/python/lib/python3.6/site-packages (from nbconver
t) (1.4.2)
Requirement already satisfied: jupyterlab-pygments in /home/jupyte
rlab/conda/envs/python/lib/python3.6/site-packages (from nbconvert
) (0.1.2)
Requirement already satisfied: pygments>=2.4.1 in /home/jupyterlab
/conda/envs/python/lib/python3.6/site-packages (from nbconvert) (2
.7.3)
Requirement already satisfied: traitlets>=4.2 in /home/jupyterlab/
conda/envs/python/lib/python3.6/site-packages (from nbconvert) (4.
3.3)
Requirement already satisfied: mistune<2,>=0.8.1 in /home/jupyterl
ab/conda/envs/python/lib/python3.6/site-packages (from nbconvert)
(0.8.4)
Requirement already satisfied: defusedxml in /home/jupyterlab/cond
a/envs/python/lib/python3.6/site-packages (from nbconvert) (0.6.0)
Requirement already satisfied: nbclient<0.6.0,>=0.5.0 in /home/jup
yterlab/conda/envs/python/lib/python3.6/site-packages (from nbconv
ert) (0.5.1)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in /home/ju
pyterlab/conda/envs/python/lib/python3.6/site-packages (from nbfor
mat>=4.4->nbconvert) (3.2.0)
Requirement already satisfied: ipython-genutils in /home/jupyterla
b/conda/envs/python/lib/python3.6/site-packages (from nbformat>=4.
4->nbconvert) (0.2.0)
Requirement already satisfied: MarkupSafe>=0.23 in /home/jupyterla
b/conda/envs/python/lib/python3.6/site-packages (from jinja2>=2.4-
>nbconvert) (1.1.1)
Requirement already satisfied: html5lib!=0.9999,!=0.99999,<0.99999
999,>=0.999 in /home/jupyterlab/conda/envs/python/lib/python3.6/si
```

```
te-packages (from bleach->nbconvert) (0.9999999)
Requirement already satisfied: six in /home/jupyterlab/conda/envs/
python/lib/python3.6/site-packages (from bleach->nbconvert) (1.15.
0)
Requirement already satisfied: decorator in /home/jupyterlab/conda
/envs/python/lib/python3.6/site-packages (from traitlets>=4.2->nbc
onvert) (4.4.2)
Requirement already satisfied: async-generator in /home/jupyterlab
/conda/envs/python/lib/python3.6/site-packages (from nbclient<0.6.
0,>=0.5.0->nbconvert) (1.10)
Requirement already satisfied: nest-asyncio in /home/jupyterlab/co
nda/envs/python/lib/python3.6/site-packages (from nbclient<0.6.0,>
=0.5.0->nbconvert) (1.4.3)
Requirement already satisfied: jupyter-client>=6.1.5 in /home/jupy
terlab/conda/envs/python/lib/python3.6/site-packages (from nbclien
t<0.6.0,>=0.5.0->nbconvert) (6.1.7)
Requirement already satisfied: attrs>=17.4.0 in /home/jupyterlab/c
onda/envs/python/lib/python3.6/site-packages (from jsonschema!=2.5
.0,>=2.4->nbformat>=4.4->nbconvert) (20.3.0)
Requirement already satisfied: setuptools in /home/jupyterlab/cond
a/envs/python/lib/python3.6/site-packages (from jsonschema!=2.5.0,
>=2.4->nbformat>=4.4->nbconvert) (49.6.0.post20201009)
Requirement already satisfied: importlib-metadata; python_version
< "3.8" in /home/jupyterlab/conda/envs/python/lib/python3.6/site-p
ackages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert) (
3.3.0)
Requirement already satisfied: pyrsistent>=0.14.0 in /home/jupyter
lab/conda/envs/python/lib/python3.6/site-packages (from jsonschema
!=2.5.0,>=2.4->nbformat>=4.4->nbconvert) (0.17.3)
Requirement already satisfied: pyzmq>=13 in /home/jupyterlab/conda
/envs/python/lib/python3.6/site-packages (from jupyter-client>=6.1
.5->nbclient<0.6.0,>=0.5.0->nbconvert) (20.0.0)
Requirement already satisfied: python-dateutil>=2.1 in /home/jupyt
erlab/conda/envs/python/lib/python3.6/site-packages (from jupyter-
client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (2.8.1)
Requirement already satisfied: tornado>=4.1 in /home/jupyterlab/co
nda/envs/python/lib/python3.6/site-packages (from jupyter-client>=
6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (6.1)
Requirement already satisfied: zipp>=0.5 in /home/jupyterlab/conda
/envs/python/lib/python3.6/site-packages (from importlib-metadata;
python_version < "3.8"->jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nb
convert) (3.4.0)
Requirement already satisfied: typing-extensions>=3.6.4; python_ve
rsion < "3.8" in /home/jupyterlab/conda/envs/python/lib/python3.6/
site-packages (from importlib-metadata; python_version < "3.8"->js
onschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert) (3.7.4.3)
```

Capstone Report

Introduction

This report is based on a python project done for the Capstone project through IBM Coursera for Machine Learning. Through my limited experience, I wanted to work with data that was provided for previously. I chose to try and find what would be considered the best neighborhood to live in specifically Toronto, Canada. For this type of project, it would normally require various public datasets for the city a person is wanting to move to. Since, I wanted to focus on Toronto I used the public venue information that is available in Foursquare API based upon Crowdsourced information.

Project Description

Finding a nice neighborhood to live in.

Most adults will eventually move to a new location whether it be for work or personal reasons. However finding a place where all of their personal needs are met can be quite the challenge. Figuring out the following can be a lot for one person to handle:

1. the average rent in a neighborhood,

2. transport amenities (both public and private) like buses and subways,

3. best forms of education in terms of schooling(if they have any, as this is a very important choice),

4. average commute time from a specific neighborhood to thework place, nearest hospitals,

5. crime rates,

6. tax rates

7. public places like parks, places to have leisure activities like going to restaurants, movie theaters, saunas, gym, etc.,

There are of course other reasons to research but the above mentioned tend to be the most common when determining a new location for living. In this project, the goal is to meet some of these requirements and guide users to choose at least 1 or more neighborhoods for their consideration during their relocation. By rating the neighborhoods based upon the available amenities they have, we can recommend the neighborhood in a city in a ranked order. This is the overall goal of this project.

Target Audience:-

Someone wanting to relocate to a new city, in this case Toronto, ON, Canada.

Stakeholders:

1. Someone who wants to relocate to a new city.

2. Myself, Toronto is a particular city that I am interested in visiting/settling in.

3. Potential businesses as well could use the information gathered from this project to determine a new location for business.

Data Description

I used public libraries and API's in this project. I used the Foursquare API, and some common Python Libraries for programming.

Foursquare API:

Foursquare provides a valuable and publically accessible location information like the ameneties in nearby locations. We as programmers use their developer tools to access the required information about the neighborhoods in a city. Using these accessed information we then rank the neighborhoods based on the ameneties they have. These services are free of charge.

I created a Foursquare developer account, and after that it provides some zip codes inside a city and for each zip code or LatLon info(Latitude and Longitude Points) the details are extracted on the amenities we expect that a neighborhood should have. The radius for this search is then set around the zip code to be around 1km.

Date Type:- JSON

Duration:-Not Available

Description of the data:-

Location coordinates obtained by Foursquare API calls.

Source:- (https://foursquare.com/ (https://foursquare.com/))

Wikipedia:

From the Wikipedia page, we can identify the neighborhood around the city. Most major cities will have this information available on Wikipedia. I accessed the web page and then extracted the neighborhood information that gave details regarding borough and neighborhood.

Date Type:- XML and HTML

Duration: N/A

Description of the data:-

Location coordinates obtained by Geocoder calls.

Source:- ([https://en.wikipedia.org/wiki/Main_Page (https://en.wikipedia.org/wiki/Main_Page)](https://en.wikipedia.org/wiki/Main_Page))

Public Programming Tools:

Some public plotting tools are used like Folium to visualize the neighborhoods in the city that we want to relocate. Then based upon the analysis of the Foursquare information, the Folium visualization can be updated to reflect the number of amenities in a neighborhood.

K-Means Clustering Algorithm on the Data:

K-Means Clustering algorithm can be used to group amenities in an area, then we can reduce the number of individual amenities comparisons to be done against each neighborhood. We can do these comparisons against the types of amenities, individually, collectively, or all together.

Data Analysis

Figuring out Latitude and Longitude

1. After learning more about the neighborhood and the details, the address of each neighborhood is prepared and Geolocator is requested for their latitude and longitude in the following code.

2. To verify the accuracy fo the latitude and the longitude values that were called, a reverse geolocation using the geocoder is done.

Result:-

There are 45 neighborhoods in Toronto. Based on the code, the neighborhoods are reduced to 1 district, North York.

Using FourSquare for Finding the public places nearby the neighborhoods

1. For the locations, the LatLon values were found, using Foursquare to find tpublic places such as coffee shops, bars, other shops, pubs, schools train startion, bus stations, parks , etc.

2. A radius of 1 km for each neighborhood within the foursquare api will return the searched public spaces and their information. It searched for at least 250 such spaces in a neighborhood.

3. The def get_category_type function analyzes the values returned by the foursquare API and filters the venue categories to find out if they match the public space types that the search was looking for.

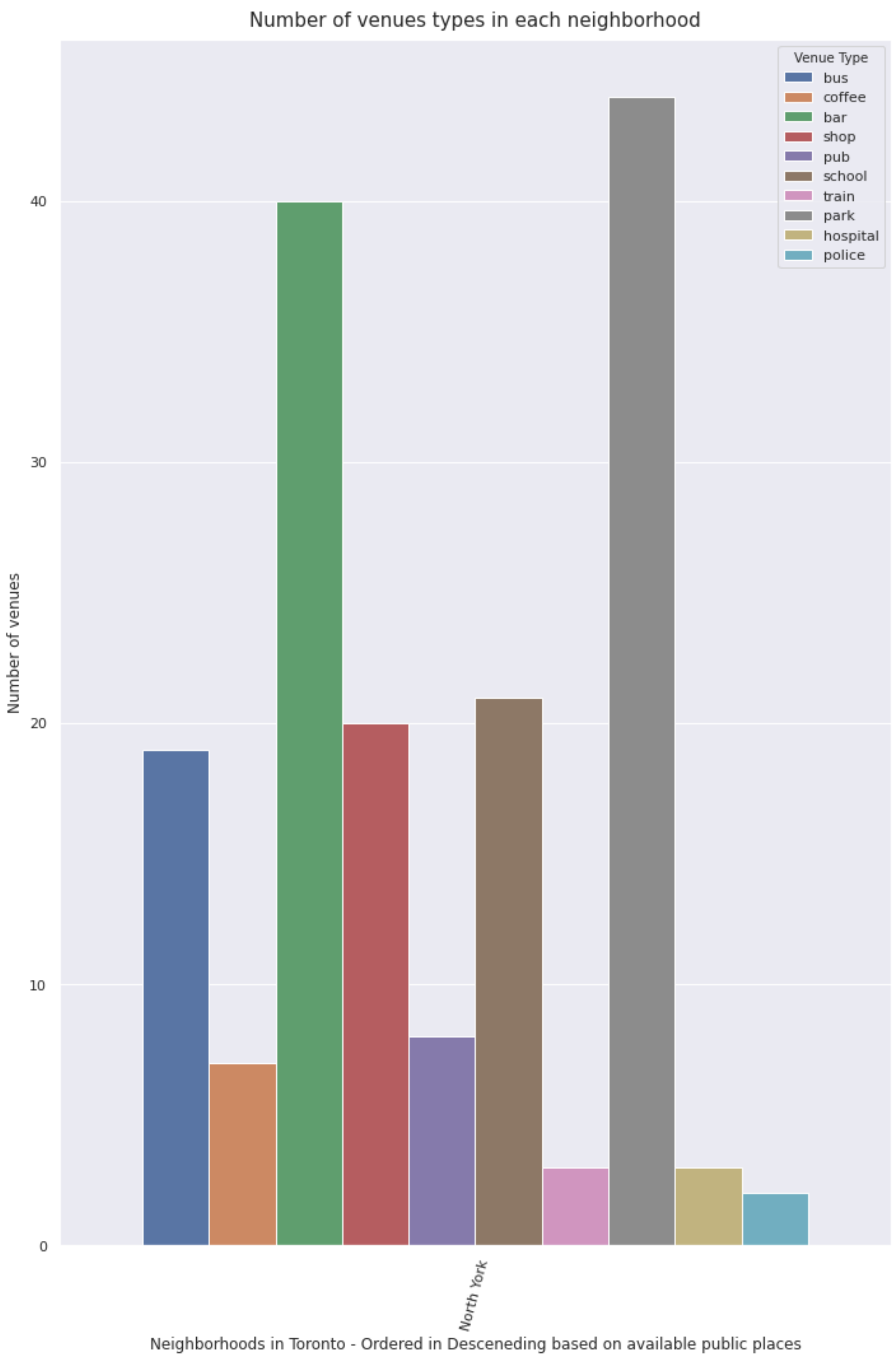
Results:-

For the 45 neighborhoods we provided to Foursquare API, we received information on 167 public places, and unusually low number considering this is a huge city.

Plots:-

The following graph Visualizes- the # of Venues obtained by each type from the FourSquare API.

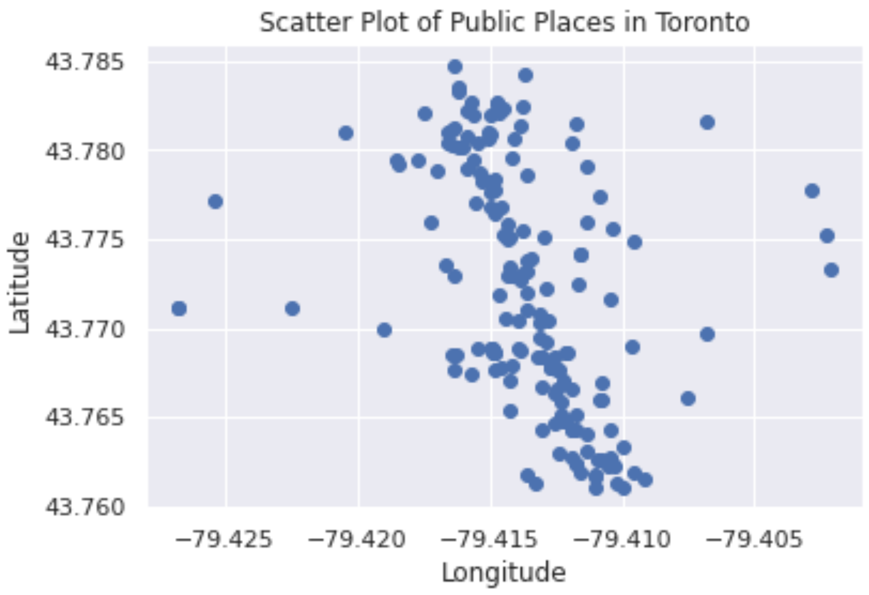
In [68]:
```
data['neighborhood'] = data['neighborhood'].apply(lambda x: x.split
(',')[0])
data.rename(columns={'query': 'Venue Type'}, inplace = True)

sns.set(style='darkgrid')
plt.figure(figsize=(10,15))
neighborhoods = data['neighborhood'].apply(lambda x: x.split(',')[0
])
sns.countplot(neighborhoods, data = data, hue = 'Venue Type', order
= neighborhoods.value_counts().index)
plt.xticks(rotation=75)
plt.title('Number of venues types in each neighborhood', pad=10, fo
ntsize = 15)
plt.ylabel('Number of venues')
plt.xlabel('Neighborhoods in Toronto - Ordered in Desceneding based
on available public places')
plt.savefig("no_of_venues.jpg")
plt.tight_layout()
```


Number of venues types in each neighborhood
Venue Type
bus
coffee
bar
shop
pub
school
train
park
hospital
police
Number of venues
40
30
20
10
0
North York
Neighborhoods in Toronto - Ordered in Desceneding based on available public places

Methods - Clustering

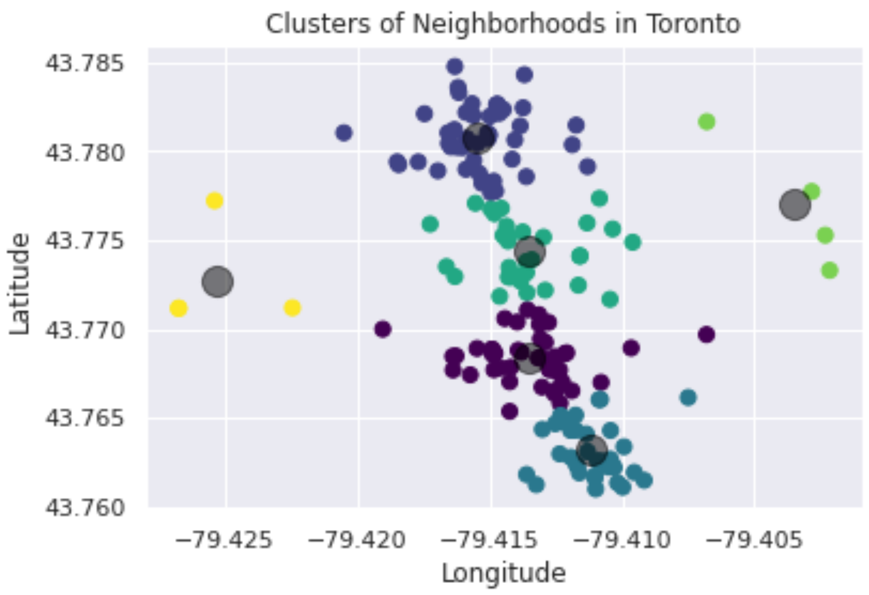
This data that was obtained from Foursquare is now analyzed further to determine what's available in North York. For more information refer the ipython notebook in the Github folder as this file you are reading.

In [69]:
```
lat_data = data['lat']
lng_data = data['lng']
neighborhood = data['neighborhood'].apply(lambda x: x.split(',')[0]
)

plt.scatter(lng_data, lat_data)
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Scatter Plot of Public Places in Toronto')
plt.savefig('Public_Places_Scatter_Plot.png')
plt.show()
```



Scatter Plot of all the venues in Toronto

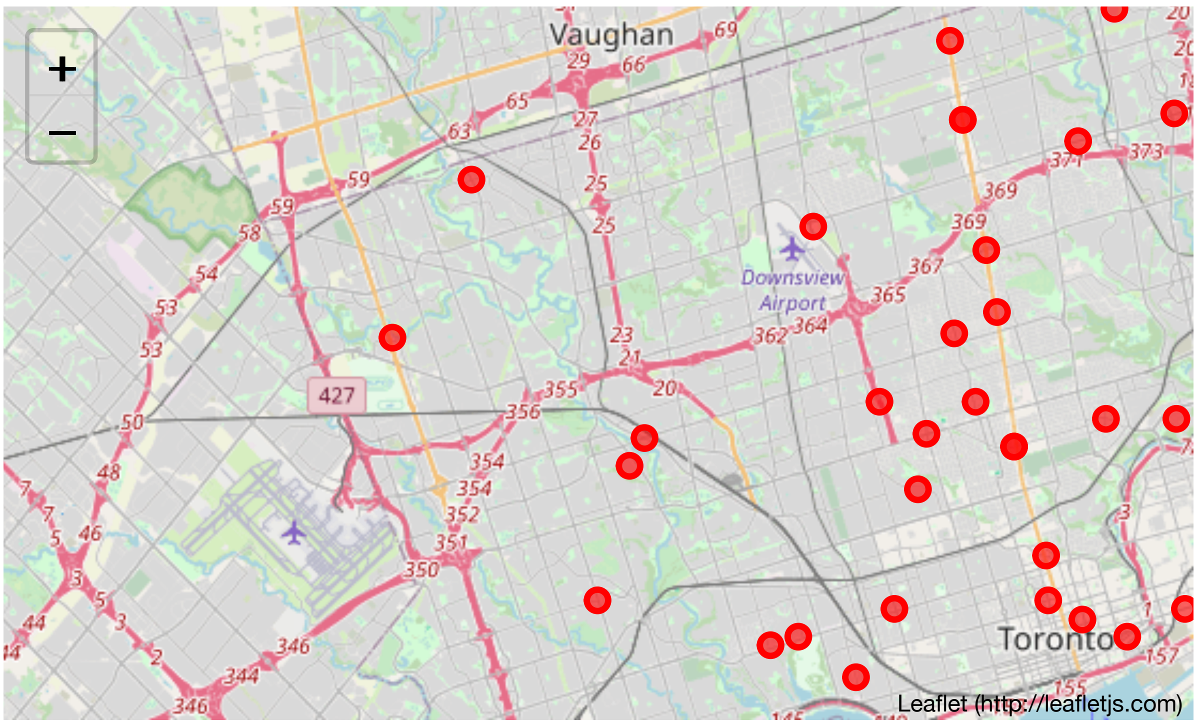
In [75]:
```
X = []
for entry in zip(lng_data, lat_data):
    X.append(list(entry))
X = np.array(X)
X
kmeans = KMeans(n_clusters=6)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)

plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='viridis')

centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0
.5);
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Clusters of Neighborhoods in Toronto')
plt.savefig('Clusters of Neighborhoods in Toronto.png')
plt.show()
```



Visualization of the Neighborhoods in Toronto (North York) Using Folium maps

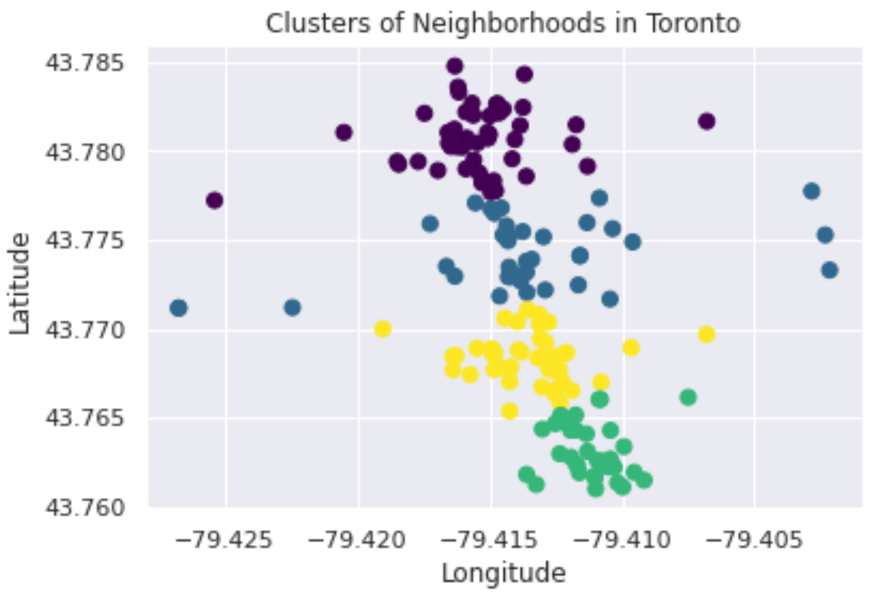
In [71]:
```
general_location = geolocator.geocode('Toronto')
geolocator = Nominatim(user_agent="coursera_capstone_project", time
out=1000)
venues_map = folium.Map(location=[43.653908, -79.384293], zoom_star
t=11)

for neighborhood, location in locations.items():
    folium.CircleMarker(
        [location[0], location[1]],
        radius=5,
        color='red',
        popup=neighborhood,
        fill = True,
        fill_color = 'red',
        fill_opacity = 0.6
    ).add_to(venues_map)

venues_map
```

Out[71]:



Visualizing public locations only in North York Neighborhood

In [76]:
```
from sklearn.metrics import pairwise_distances_argmin

def find_clusters(X, n_clusters, rseed=2):
    # 1. Randomly choose clusters
    rng = np.random.RandomState(rseed)
    i = rng.permutation(X.shape[0])[:n_clusters]
    centers = X[i]

    while True:
        # 2a. Assign labels based on closest center
        labels = pairwise_distances_argmin(X, centers)

        # 2b. Find new centers from means of points
        new_centers = np.array([X[labels == i].mean(0)
                                for i in range(n_clusters)])

        # 2c. Check for convergence
        if np.all(centers == new_centers):
            break
        centers = new_centers

    return centers, labels

centers, labels = find_clusters(X, 4)
plt.scatter(X[:, 0], X[:, 1], c=labels,
            s=50, cmap='viridis');
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Clusters of Neighborhoods in Toronto')
plt.savefig('Clusters of Neighborhoods in Toronto Reduced clusters.
png')
plt.show()
```



Result

Based on the data provided, North York had the most amenities provided when compared to the remaining neighborhoods. So when a person wants to relocate based upon all the avilable public amenities they can recognize that North York is the best of all the available options. This conclusion is solely based on the quantity of the public amenities available, not quality of service. This is a drawback of the above project in addition to the lack of public amenities being showcased in other boroughs.

Discussion

This project can be modified for any major city and used to determine a suitable neighborhood to live in. However, choosing a place to live in the future just based on the number of amenities does not equal to a perfect place. Other factors need to be taken into consideration such as peace of mind and health. The quality of services as well as the other factors should be taken into consideration, further analysis would need to be conducted. There were some troubles with this project because of the number of calls could be made daily with Foursquare API 4. Changing to Google Places API perhaps would be more suitable because it would increase from 950 to 10,000 calls per day.

Conclusion

This project was a bit challenging to work on because the numbers that were expected were not being received. However, I think it was a great opportunity to practice and understand geolocation and the use of APIs better. I have a better understanding of how to rank based on quantity, neighborhoods so that in the future I would be able to assist anyone in relocating.

References

1. https://en.wikipedia.org/wiki/Main_Page (https://en.wikipedia.org/wiki/Main_Page)

2. https://foursquare.com (https://foursquare.com)

3. https://jakevdp.github.io/PythonDataScienceHandbook/05.11-k-means.html (https://jakevdp.github.io/PythonDataScienceHandbook/05.11-k-means.html)

4. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html (https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html)

5. https://en.wikipedia.org/wiki/Toronto (https://en.wikipedia.org/wiki/Toronto)

6. https://en.wikipedia.org/wiki/List_of_neighbourhoods_in_Toronto (https://en.wikipedia.org/wiki/List_of_neighbourhoods_in_Toronto)

7. https://notebook.community/mohanprasath/Course-Work/coursera/applied_data_science_capstone/Capstone%20Report (https://notebook.community/mohanprasath/Course-Work/coursera/applied_data_science_capstone/Capstone%20Report)

In [ ]:

```
```