Project Capstone: Battle of Neighbourhoods - Week 1

1. Introduction/Business Problem

Toronto is the capital city of the Canadian province of Ontario. With a recorded population of 2,731,571 in 2016, It is the most populous city in Canada and the fourth most populous city in North America. The city is the anchor of the Golden Horseshoe, an urban agglomeration of 9,245,438 people (as of 2016) surrounding the western end of Lake Ontario, while the Greater Toronto Area (GTA) proper had a 2016 population of 6,417,516. Toronto is an international centre of business, finance, arts, and culture, and is recognized as one of the most multicultural and cosmopolitan cities in the world the diverse population of Toronto reflects its current and historical role as an important destination for immigrants to Canada. More than 50 percent of residents belong to a visible minority population group, and over 200 distinct ethnic origins are represented among its inhabitants. While the majority of Torontonians speak English as their primary language, over 160 languages are spoken in the city

2. Business Problem

Coming down to business problems, There is any a hotel/restaurant near beach side. As it is a famous tourist spot, there is already lots of attention towards it. I need to search the hotel near to the city where more people are attracted and comfortable for a stay and with a good meal. Unfortunately Only the five star hotels provide the variants menus according to the taste foreign camesfrom. The menu foods can be variants with Italian, American, typical south & north, asian foods and indian foods to grab their taste.

3. Expected / Interested Audience

20% local and 80% foreign peoples visit Toronto once in a year. Some people stay for couple of days or more.Also they find some place for hangout or a good meal.Their main focus might be belonging to stay somewhere near to reach venues.

4. Data Collection

To complete this case, we require data set as follows:

Part_1:

- Venue data relate to Hotel & Restaurant. This will help us find the neighborhoods that are more suitable to find the best hotel & Restaurant Part_2:
- List of neighborhoods in Toronto, Ontario Canada
- Latitute and Longiture of these neigborhoods

5. Extracting The Data

Scrapping of Toronto neighborhoods via Wikipedia Getting Latitute and Longitude data of these neighborhoods via Geocoder packages

Part 1 - Import Library

```
In [1]: # Import libraries
        import numpy as np # data in a vectorized manner manipulation
        import pandas as pd # data analsysis
        import requests # HTTP library
        from bs4 import BeautifulSoup # scraping library

        from pandas.io.json import json_normalize # tranform JSON file into a pandas data
        import json # JSON files manipulation

        from sklearn.cluster import KMeans # clustering algorithm

        pd.set_option('display.max_columns', None)
        pd.set_option('display.max_rows', None)

        # Matplotlib and associated plotting modules
        import matplotlib.cm as cm
        import matplotlib.colors as colors
        import matplotlib.pyplot as plt

        #!conda install -c conda-forge geopy --yes
        from geopy.geocoders import Nominatim

        #!conda install -c conda-forge folium=0.5.0 --yes
        import folium # map rendering library

        print("@@Libraries Imported@@")
```

```
@@Libraries Imported@@
```

foursquare credential id

```
In [2]: CLIENT_ID = 'Y4AOTAYDY23ILH3DR4XZZDYEZDHNECVYXORDA4N4WWLLJ3NB' # your Foursquare
        CLIENT_SECRET = 'XSI4HLFPMXC1PFUWGYR1U35DVKVOO32KWR3FQGGIJINNPEG2' # your Foursqu
        VERSION = '20200804' # Foursquare API version
        LIMIT = 100

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

```
Your credentails:
CLIENT_ID: Y4AOTAYDY23ILH3DR4XZZDYEZDHNECVYXORDA4N4WWLLJ3NB
CLIENT_SECRET:XSI4HLFPMXC1PFUWGYR1U35DVKVOO32KWR3FQGGIJINNPEG2
```

Credentials and Location

Use geopy library to get the latitude and longitude values of Toronto

In [3]:
```
address = 'Toronto, Ontario'

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

```
The geograpical coordinate of Toronto are 43.6534817, -79.3839347.
```

#### Search for hotel & restaurant within 1000 Meter

In [5]:
```
#Quering for hotel & restaurant

qry_htl = 'hotel'
qry_rst = 'restaurant'

radius = 1000
url_hotel = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secr
url_restaurant = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_
#url
```

Send the GET Request of hotel & restaurants and examine the results

In [6]:
```
results_hotel = requests.get(url_hotel).json()
results_restaurant = requests.get(url_restaurant).json()
results_restaurant
```

Out[6]:

```
{'meta': {'code': 200, 'requestId': '5f31251970412b0a532df117'},
 'response': {'venues': [{'categories': [{'icon': {'prefix': 'https://ss3.4s
qi.net/img/categories_v2/food/default_',
       'suffix': '.png'},
      'id': '4bf58dd8d48988d14e941735',
      'name': 'American Restaurant',
      'pluralName': 'American Restaurants',
      'primary': True,
      'shortName': 'American'}],
    'hasPerk': False,
    'id': '4ad4c05ff964a52048f720e3',
    'location': {'address': '110 Chestnut Street',
     'cc': 'CA',
     'city': 'Toronto',
     'country': 'Canada',
     'distance': 224,
     'formattedAddress': ['110 Chestnut Street',
      'Toronto ON M5G 1R3',
      'Canada'],
     'labeledLatLngs': [{'label': 'display'
```

#### Get relevant part of JSON and transform it into a pandas dataframes

In [7]:

```
# assign relevant part of JSON to venues
venues_hotel      = results_hotel['response']['venues']
venues_restaurant = results_restaurant['response']['venues']

# tranform venues into a dataframe and merging both data
df_hotel      = json_normalize(venues_hotel)
df_restaurant = json_normalize(venues_restaurant)

df = pd.concat([df_hotel, df_restaurant])

print("There are {} restaurants and hotels at Toronto".format(df.shape[0]))
```

```
There are 100 restaurants and hotels at Toronto
```

Define information of interest and filter dataframe

In [8]:

```
# keep only columns that include venue name, and anything that is associated with
flt_columns = ['name', 'categories'] + [col for col in df.columns if col.startswi
df_flt      = df.loc[:, flt_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
df_flt['categories'] = df_flt.apply(get_category_type, axis=1)

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

#dataframe_filtered
hotels_df=df_flt[['name','categories','distance','lat','lng','id']]
hotels_df.head()
```

Out[8]:

	name	categories	distance	lat	lng	id
0	Sheraton Centre Toronto Hotel	Hotel	324	43.650594	-79.384530	4ab2d511f964a5209b6c20e3
1	One King West Hotel & Residence	Hotel	686	43.649139	-79.377876	4af96fbbf964a520c01122e3
2	Chelsea Hotel	Hotel	562	43.658498	-79.383097	51d212c3498ebf27dc469bc9
3	The Omni King Edward Hotel	Hotel	797	43.649191	-79.376006	4adf7d0bf964a520127b21e3
4	Pantages Hotel & Spa	Hotel	410	43.654498	-79.379035	4ae61cf6f964a520caa421e3

Location of Hotels

In [13]:

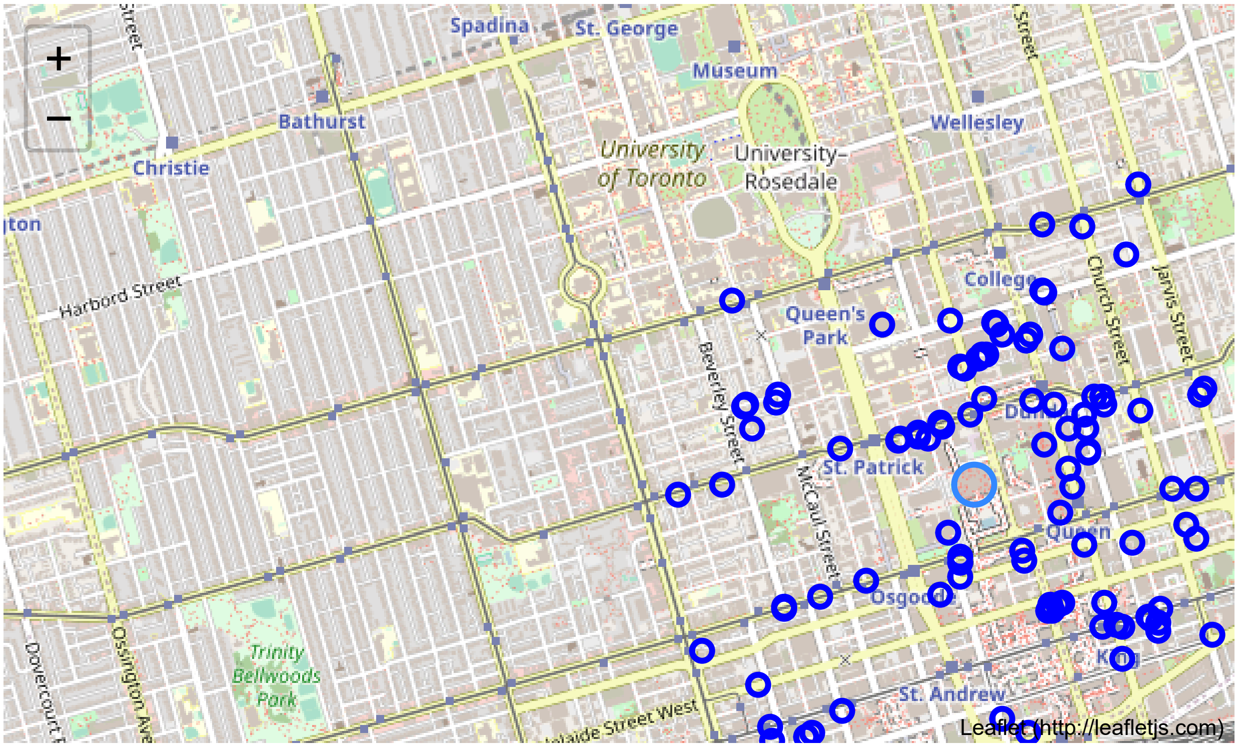
```
hotels_map = folium.Map(location=[latitude, longitude], zoom_start=11) # generate

# add a red circle marker to represent the core location of Toronto
folium.features.CircleMarker(
    [latitude, longitude],
    radius=10,
    color='#3388ff',
    popup='Toronto',
    fill_opacity = 0.7
).add_to(hotels_map)

# add the Italian restaurants as blue circle markers
for lat, lng, label in zip(hotels_df.lat, hotels_df.lng, hotels_df.name):
    folium.features.CircleMarker(
        [lat, lng],
        radius=5,
        color='Blue',
        fill_opacity=0.7
    ).add_to(hotels_map)

# display map
hotels_map
```

Out[13]:



In [ ]: