CAPSTONE PROJECT

The idea of this project is to provide the best location for a restaurant based on external sources of data. What I will try during this notebook is to show different sources of data to identify the best location.

DATASET USED

1.- Foursquare info from previous week

2.- Neiborhoud boundaries from (https://open.toronto.ca/dataset/neighbourhoods/ (https://open.toronto.ca/dataset/neighbourhoods/))

3.- Business Improvement areas (https://open.toronto.ca/dataset/business-improvement-areas/ (https://open.toronto.ca/dataset/business-improvement-areas/))

During this notebook I will try to link the situation of the main food related placed in the city of toronto with the biggest business development area. This will lead us to find which is the % of restaurantes in each area and the proportion compared to the rest. Based on this if we want to place a restaurant it should be done in the best business area with the lowest restaurant rate

METHODOLOGY

I provide an study where I evaluate the realtionship between the number of elements in each area compared with the number of food related ones. Lowest ratio is the indicator to place the restaurant

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.cm as cm
from scipy.spatial import distance_matrix
import matplotlib.colors as colors
import folium # plotting library
from sklearn.cluster import KMeans
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values
from math import cos, asin, sqrt
%matplotlib inline
```

READ Datasets

In [2]:

```
# Foursquare data:
df_square=pd.read_csv('Toronto_data.csv')
df_square.head()
```

Out[2]:

	Unnamed: 0	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	0	Rouge	43.806686	-79.194353	Wendy's	43.807448	-79.199056	Fast Food Restaurant
1	1	Malvern	43.806686	-79.194353	Wendy's	43.807448	-79.199056	Fast Food Restaurant
2	2	Highland Creek	43.784535	-79.160497	Royal Canadian Legion	43.782533	-79.163085	Bar
3	3	Highland Creek	43.784535	-79.160497	Affordable Toronto Movers	43.787919	-79.162977	Moving Target
4	4	Rouge Hill	43.784535	-79.160497	Royal Canadian Legion	43.782533	-79.163085	Bar

In [3]:

```
df_areas=pd.read_csv('Business Improvement Areas Data.csv')
df_areas.head()
```

Out[3]:

	_id	AREA_ID	DATE_EFFECTIVE	AREA_ATTR_ID	PARENT_AREA_ID	AREA_SHORT_CODE	AREA_LONG_CODE	A
0	739	2478937	2019-05-28T21:47:59	26004921	NaN	020-01	020-01	
1	740	2478936	2019-05-28T21:47:59	26004920	NaN	042-01	042-01	Li
2	741	2478935	2019-05-28T21:47:59	26004919	NaN	093-01	093-01	
3	742	2478934	2019-05-28T21:47:59	26004918	NaN	033-00	033-00	
4	743	2478933	2019-05-28T21:47:59	26004917	NaN	002-00	002-00	

In the code below we will assign the closest business area based on distance to the center of the area. This way we can calculate the total number of places by AREA, which will give us a size of it.

In [4]:

```
def distance(lat1, lon1, lat2, lon2):
    p = 0.017453292519943295
    a = 0.5 - cos((lat2-lat1)*p)/2 + cos(lat1*p)*cos(lat2*p) * (1-cos((lon2-lon1)*p)) / 2
    return 12742 * asin(sqrt(a))

def closest(data, v):
    return min(data, key=lambda p: distance(v['LATITUDE'],v['LONGITUDE'],p['LATITUDE'],p['
LONGITUDE']))['AREA_NAME']








def find_area():
    tempData = []
    for index, row in df_areas.iterrows():
        tempDict = {}
        tempDict['LATITUDE']=row['LATITUDE']
        tempDict['LONGITUDE']=row['LONGITUDE']
        tempDict['AREA_NAME']=row['AREA_NAME']
        tempData.append(tempDict)
    return_value=[]
    for index, row in df_square.iterrows():
        temp_results = {}
        tempRow = {'LATITUDE': row['Venue Latitude'], 'LONGITUDE': row['Venue Longitude']}
        temp_results['AREA']=closest(tempData,tempRow)
        temp_results['Venue Category']=row['Venue Category']
        temp_results['Venue Latitude']=row['Venue Latitude']
        temp_results['Venue Longitude']=row['Venue Longitude']
        return_value.append(temp_results)
    return return_value

df_square['AREA']=""
df_temp_rest=find_area()


df = pd.DataFrame(df_temp_rest, columns =['AREA', 'Venue Category' ,'Venue Latitude','Venu
e Longitude' ])

data_grouped=df.groupby("AREA")["AREA"].count()

df_n = pd.DataFrame(data_grouped, columns=['AREA'])

df_n.rename(columns={'AREA':'Total'},inplace=True)

df_population=df_n.sort_values(by=['Total'],ascending=False)
```

First let's find out where are the actual restaurants placed

In [5]:

```
df.loc[df['Venue Category'].str.contains("Restaurant"), 'food_related'] = True
df.loc[df['Venue Category'].str.contains("Gastropub"), 'food_related'] = True
#df_square.loc[df_square['Venue Category'].str.contains("Bar"), 'food_related'] = True


df_area_food=df[df['food_related'] == True]
len(df_area_food)

df_area_grouped=df_area_food.groupby("AREA")["AREA"].count()
df_area_grouped = pd.DataFrame(df_area_grouped, columns =['AREA'])
df_area_grouped.rename(columns={'AREA':'Total'},inplace=True)

df_restaurants=df_area_grouped.sort_values(by=['Total'],ascending=False)
```

Let's find out which is the best business area based on the propotion of restaurants

In [6]:

```
# Number of Items
df_population

# Number of Restaurants
df_restaurants

df_test=df_population.join(df_restaurants, lsuffix='_caller', rsuffix='_other')

df_test['Ratio']=(df_test['Total_other']*100)/df_test['Total_caller']

df_test=df_test.reset_index()

df_test.head()
```

Out[6]:

	AREA	Total_caller	Total_other	Ratio
0	Financial District	976	262.0	26.844262
1	Downtown Yonge	305	65.0	21.311475
2	Toronto Entertainment District	249	38.0	15.261044
3	Kennedy Road	204	59.0	28.921569
4	Kensington Market	199	60.0	30.150754

In [7]:

```
address = 'Toronto'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Toronto are {}, {}.'.format(latitude, longitude))
map_toronto = folium.Map(location=[latitude, longitude], zoom_start=10)
```

```
The geograpical coordinate of Toronto are 43.653963, -79.387207.
```

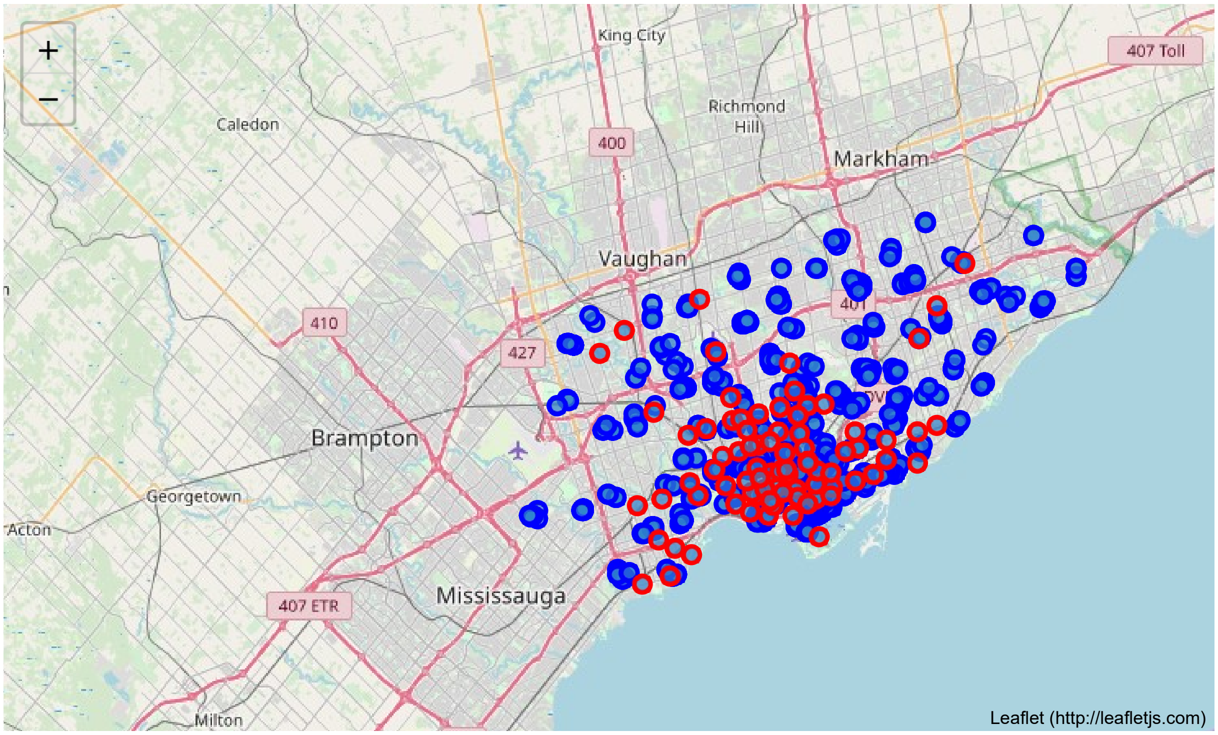
In [8]:

```
for lat, lng, venue_type in zip(df['Venue Latitude'], df['Venue Longitude'], df['Venue Cat
egory']):
    label = '{}'.format(venue_type)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_toronto)

for lat, lng, area_name in zip(df_areas['LATITUDE'], df_areas['LONGITUDE'], df_areas['AREA
_NAME']):
    label = '{}'.format(area_name)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='red',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_toronto)

map_toronto
```

Out[8]:


+
−
King City
407 Toll
Richmond Hill
Caledon
400
Markham
Vaughan
401
410
427
Brampton
Georgetown
Acton
407 ETR
Mississauga
Milton
Leaflet (http://leafletjs.com)