Report: Capstone Project - The Battle of Neighborhoods

Introduction

Business problem: find the most profitable location to open a pizzeria in Scarborough, Toronto, Canada

Interested parties: Luigi and his family, who would like to open a pizzeria near their house in Scarborough.

Data

Raw data present postal codes, neighborhoods and geographical location of Scarborough. The data source is https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M and geographical data is sourced from geopy.geocoders.

Notebooks showing the code to create the following table were created in week 3 of this course and can be found under following links:

- https://gist.github.com/GruenscS/cfe8a2daf32499380e033bd2f7c756e3
- https://gist.github.com/GruenscS/8738544df5961b0b4ba2e97c5d3c6c6d

	PostalCode	Borough	Neighborhood	Latitude	Longitude
0	M1B	Scarborough	Rouge, Malvern	43.811650	-79.195561
1	M1C	Scarborough	Highland Creek, Rouge Hill, Port Union	43.785605	-79.158701
2	M1E	Scarborough	Guildwood, Morningside, West Hill	43.765690	-79.175299
3	M1G	Scarborough	Woburn	43.768216	-79.217610
4	M1H	Scarborough	Cedarbrae	43.769608	-79.239440
5	M1J	Scarborough	Scarborough Village	43.743085	-79.232172
6	M1K	Scarborough	East Birchmount Park, Ionview, Kennedy Park	43.726260	-79.263670
7	M1L	Scarborough	Clairlea, Golden Mile, Oakridge	43.713213	-79.284910
8	M1M	Scarborough	Cliffcrest, Cliffside, Scarborough Village West	43.723575	-79.234976
9	M1N	Scarborough	Birch Cliff, Cliffside West	43.696690	-79.260069
10	M1P	Scarborough	Dorset Park, Scarborough Town Centre, Wexford ...	43.759975	-79.268974
11	M1R	Scarborough	Maryvale, Wexford	43.750803	-79.300560
12	M1S	Scarborough	Agincourt	43.793940	-79.267976
13	M1T	Scarborough	Clarks Corners, Sullivan, Tam O'Shanter	43.784725	-79.299244
14	M1V	Scarborough	Agincourt North, L'Amoreaux East, Milliken, St...	43.817595	-79.280147
15	M1W	Scarborough	L'Amoreaux West, Steeles West	43.800698	-79.320740
16	M1X	Scarborough	Upper Rouge	43.834215	-79.216701

Methodology

Following libraries were installed for data analysis:

```
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analsysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

!conda install -c conda-forge geopy --yes
# convert an address into latitude and longitude values
from geopy.geocoders import Nominatim

import requests # library to handle requests
# tranform JSON file into a pandas dataframe
from pandas.io.json import json_normalize

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

# import k-means from clustering stage
from sklearn.cluster import KMeans

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

Neighborhoods in Scarborough where explored using Foursquare API with following URL and Foursquare API version:

url: 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&ll={},{}&v={}&radius={}&limit={}'.format (CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, radius, LIMIT)

version: '20180605'

A function to repeat the exploring process to all neighborhoods in Scarborough was created:

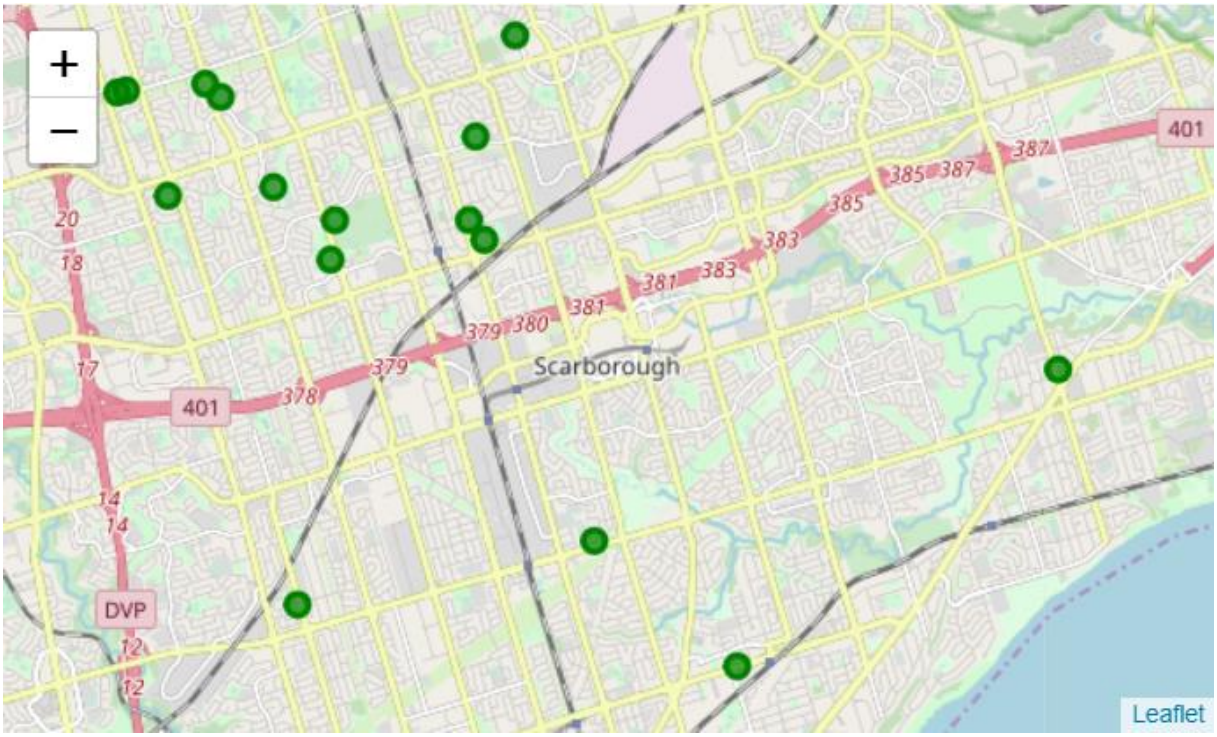
```
import urllib
def getNearbyVenues(names, latitudes, longitudes, radius=5000, category
Ids=''):
    try:
        venues_list=[]
        for name, lat, lng in zip(names, latitudes, longitudes):
            #print(name)

            # create the API request URL
            url = 'https://api.foursquare.com/v2/venues/search?&client_
id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(CLIENT_
ID, CLIENT_SECRET, VERSION, lat, lng, radius, LIMIT)

            if (categoryIds != ''):
                url = url + '&categoryId={}'
                url = url.format(categoryIds)

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

            # return only relevant information for each nearby venue
            for v in results:
                success = False
                try:
                    category = v['categories'][0]['name']
                    success = True
                except:
                    pass

                if success:
                    venues_list.append([(
                        name,
                        lat,
                        lng,
                        v['name'],
                        v['location']['lat'],
                        v['location']['lng'],
                        v['categories'][0]['name']
                    )])

        nearby_venues = pd.DataFrame([item for venue_list in venues_lis
t for item in venue_list])
        nearby_venues.columns = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    except:
        print(url)
        print(response)
        print(results)
        print(nearby_venues)

    return(nearby_venues)
```

Following approaches were done to find the most profitable location of a pizzeria:

```
# Use category id 4bf58dd8d48988d1ca941735 to only find pizzerias in Scarborough
toronto_pizza = getNearbyVenues(names=df['Neighborhood'],
                                latitudes=df['Latitude'],
                                longitudes=df['Longitude'],
                               radius=1000, categoryIds='4bf58dd8d48988
d1ca941735')
```

 # Use category id 4bf58dd8d48988d13d941735 to only find high schools in Scarborough, since students are good customers

```
toronto_venues_highschools = getNearbyVenues(names=df['Neighborhood'],
                                             latitudes=df['Latitude'],
                                             longitudes=df['Longitude'], radius=1
000, categoryIds='4bf58dd8d48988d13d941735')
```

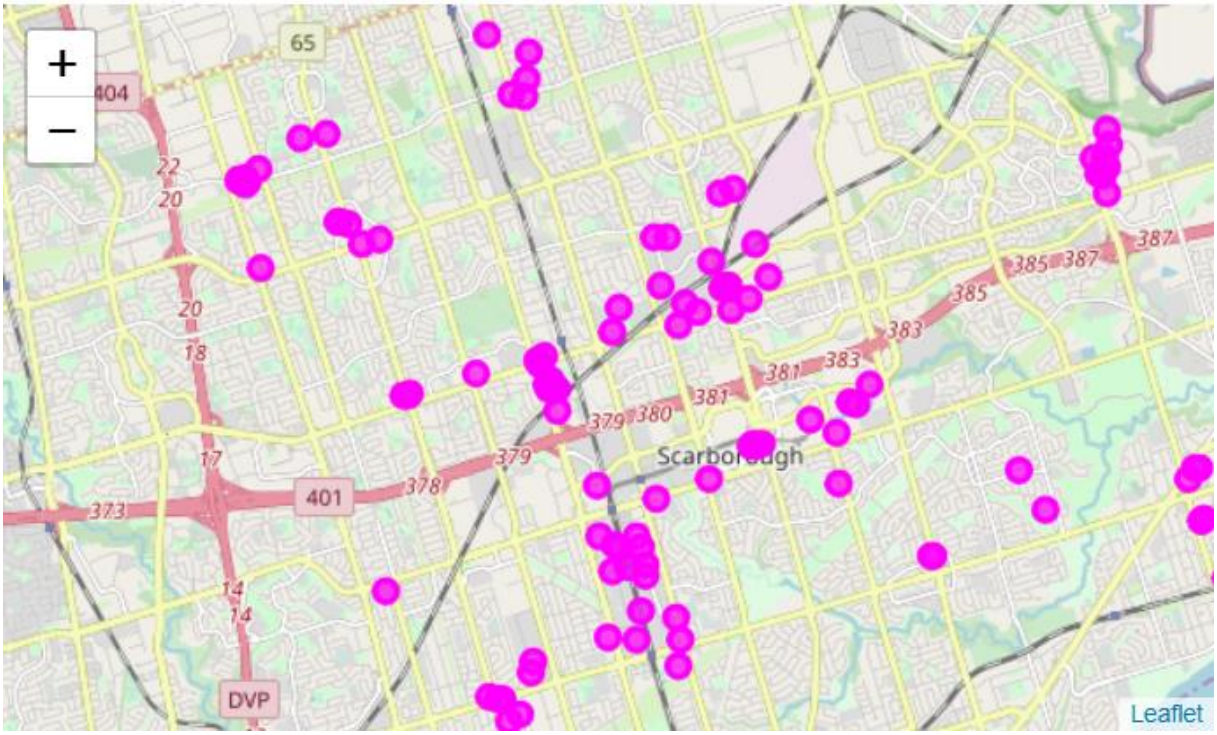


 # Use category id 4bf58dd8d48988d1ae941735 to only find universities in Scarborough, since uni students are good customers. ***No universities in Scarborough were found.***

```
toronto_venues_uni = getNearbyVenues(names=df['Neighborhood'],
                                             latitudes=df['Latitude'],
                                             longitudes=df['Longitude'], radius=1
000, categoryIds='4bf58dd8d48988d1ae941735')
```

 # Use category id 4bf58dd8d48988d124941735 to only find offices in Scarborough, since employees are good customers.

```
toronto_venues_office = getNearbyVenues(names=df['Neighborhood'],
                                        latitudes=df['Latitude'],
                                        longitudes=df['Longitude'], radius=1
000, categoryIds='4bf58dd8d48988d124941735')
```



 # Add data of Pizza, High Schools and Offices to table of neighborhoods in Scarborough:

```
def addColumn(startDf, columnTitle, dataDf):
    grouped = dataDf.groupby('Neighborhood').count()

    for n in startDf['Neighborhood']:
        try:
            startDf.loc[startDf['Neighborhood'] == n,columnTitle] = gro
uped.loc[n, 'Venue']
        except:
            startDf.loc[startDf['Neighborhood'] == n,columnTitle] = 0

df_data = df.copy()
addColumn(df_data, 'Pizza', toronto_pizza)
addColumn(df_data, 'High Schools', toronto_venues_highschools)
addColumn(df_data, 'Offices', toronto_venues_office)
```

	PostalCode	Borough	Neighborhood	Latitude	Longitude	Pizza	High Schools	Offices
0	M1B	Scarborough	Rouge, Malvern	43.811650	-79.195561	1.0	0.0	8.0
1	M1C	Scarborough	Highland Creek, Rouge Hill, Port Union	43.785605	-79.158701	0.0	0.0	4.0
2	M1E	Scarborough	Guildwood, Morningside, West Hill	43.765690	-79.175299	2.0	1.0	11.0
3	M1G	Scarborough	Woburn	43.768216	-79.217610	5.0	0.0	4.0
4	M1H	Scarborough	Cedarbrae	43.769608	-79.239440	3.0	0.0	10.0
5	M1J	Scarborough	Scarborough Village	43.743085	-79.232172	3.0	1.0	0.0
6	M1K	Scarborough	East Birchmount Park, Ionview, Kennedy Park	43.726260	-79.263670	3.0	1.0	16.0
7	M1L	Scarborough	Clairlea, Golden Mile, Oakridge	43.713213	-79.284910	2.0	1.0	7.0
8	M1M	Scarborough	Cliffcrest, Cliffside, Scarborough Village West	43.723575	-79.234976	2.0	2.0	6.0
9	M1N	Scarborough	Birch Cliff, Cliffside West	43.696690	-79.260069	1.0	1.0	4.0
10	M1P	Scarborough	Dorset Park, Scarborough Town Centre, Wexford ...	43.759975	-79.268974	2.0	1.0	18.0
11	M1R	Scarborough	Maryvale, Wexford	43.750803	-79.300560	4.0	1.0	8.0
12	M1S	Scarborough	Agincourt	43.793940	-79.267976	2.0	3.0	19.0
13	M1T	Scarborough	Clarks Corners, Sullivan, Tam O'Shanter	43.784725	-79.299244	3.0	3.0	14.0
14	M1V	Scarborough	Agincourt North, L'Amoreaux East, Milliken, St...	43.817595	-79.280147	2.0	1.0	5.0
15	M1W	Scarborough	L'Amoreaux West, Steeles West	43.800698	-79.320740	3.0	5.0	17.0
16	M1X	Scarborough	Upper Rouge	43.834215	-79.216701	0.0	0.0	1.0

Number of pizzerias, high schools and offices were summed up per neighborhood and weighted according to the following sheme:

```
# negative weight, because Luigi wants to open a pizzeria
#and thus wants to avoid concurrence as much as possible
weight_pizza = -1

# positive weight, because high school students are good customers
weight_schools = 1

# positive weight because employees are even better customers
weight_offices = 2
```

#Sums of each category were multiplied with weights to generate following table, which summarizes the best locations in Scarborough to open a pizzeria

```
df_weighted['Score'] = df_data['Pizza'] * weight_pizza + df_data['High Schools'] * weight_schools + df_data['Offices'] * weight_offices
df_weighted = df_weighted.sort_values(by=['Score'], ascending=False)
df_weighted
```

Category IDs were used from Foursquare url: https://developer.foursquare.com/docs/resources/categories

Results

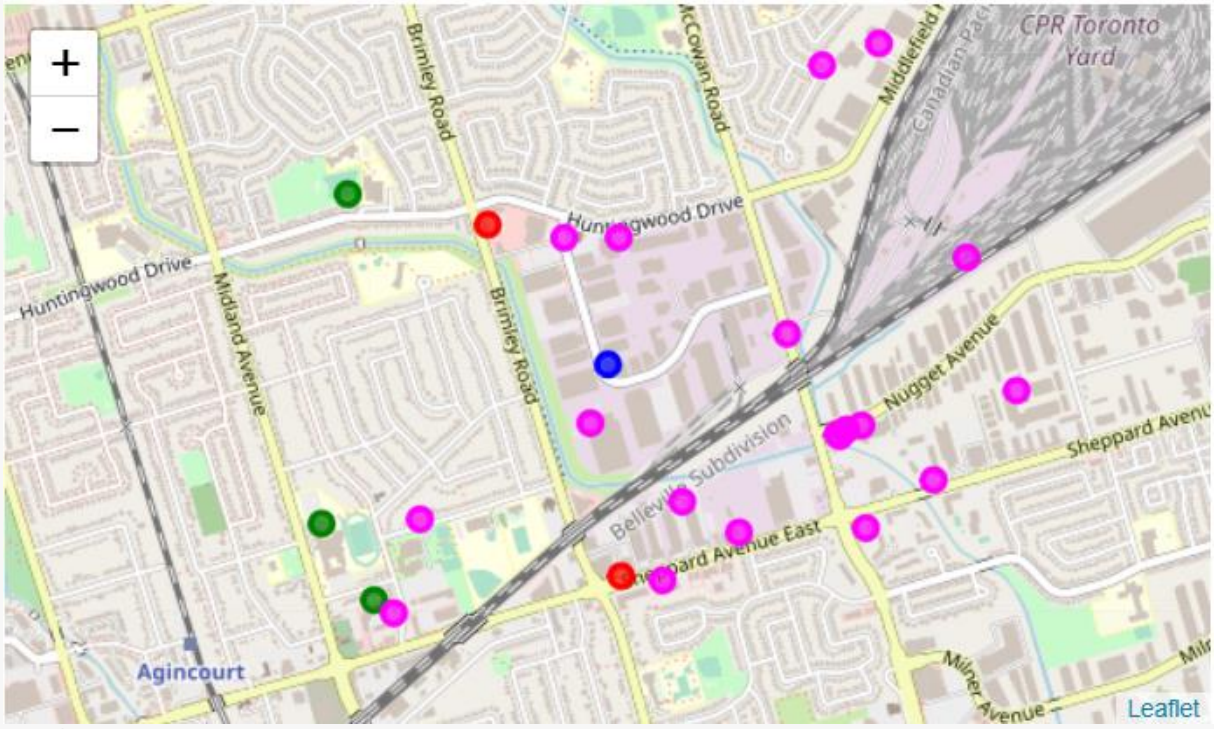
Sums of each category were multiplied with weights to generate following table, which summarizes the best locations in Scarborough to open a pizzeria. Agincourt presents the highest score of neighborhoods in Scarborough.

	Neighborhood	Score
12	Agincourt	39.0
15	L'Amoreaux West, Steeles West	36.0
10	Dorset Park, Scarborough Town Centre, Wexford ...	35.0
6	East Birchmount Park, Ionview, Kennedy Park	30.0
13	Clarks Corners, Sullivan, Tam O'Shanter	28.0
2	Guildwood, Morningside, West Hill	21.0
4	Cedarbrae	17.0
0	Rouge, Malvern	15.0
7	Clairlea, Golden Mile, Oakridge	13.0
11	Maryvale, Wexford	13.0
8	Cliffcrest, Cliffside, Scarborough Village West	12.0
14	Agincourt North, L'Amoreaux East, Milliken, St...	9.0
9	Birch Cliff, Cliffside West	8.0
1	Highland Creek, Rouge Hill, Port Union	8.0
3	Woburn	3.0
16	Upper Rouge	2.0
5	Scarborough Village	-2.0

```
pizza_win = df[df['Neighborhood'] == 'Agincourt']

pizza_win
```

	PostalCode	Borough	Neighborhood	Latitude	Longitude
12	M1S	Scarborough	Agincourt	43.79394	-79.267976

Following maps shows the neighborhood of Agincourt (blue) with existing pizzerias (red), high schools (green) and offices (fuchsia).




```
latitude = 43.793940
longitude = -79.267976

# create map of Toronto using latitude and longitude values

map_pizza_result = folium.Map(location=[latitude, longitude], zoom_star
t=14)

# add markers to map

for lat, lng, local in zip(pizza_win['Latitude'], pizza_win['Longitude'
], pizza_win['Neighborhood']):
    label = '{}'.format(local)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='blue',
        fill_opacity=0.7).add_to(map_pizza_result)

addToMap(toronto_pizza[toronto_pizza['Neighborhood'] == 'Agincourt'], '
red', map_pizza_result)
addToMap(toronto_venues_highschools[toronto_venues_highschools['Neighbo
rhood'] == 'Agincourt'], 'green', map_pizza_result)
addToMap(toronto_venues_office[toronto_venues_office['Neighborhood'] ==
'Agincourt'], 'fuchsia', map_pizza_result)

map_pizza_result
```

Discussion

Further analysis can be done on the second and third ranking neighborhood in Scarborough to find alternative spots that might be lucrative but also more close to Luigis home.

The data analysis can be improved with following extensions:

- Consider more categories, e.g. Nightlife Spots (4d4b7105d754a06376d81259) or Food/Restaurants (4d4b7105d754a06374d81259), which can be a source of customers but also present some concurrence.

Conclusion section where you conclude the report.

Agincourt is the best option to open a pizzeria for Luigi.