IBM Data Science Professional Certification

Capstone Project From IBM/Coursera

Himanshi Sharma

Capstone Project - The Battle of the Neighborhoods

Applied data science capstone Project

Author: Himanshi Sharma

Date: 17 August 2020

Table of contents

1.Introduction: Business Problem

In this project we will try to find an location for a restaurant.

Since there are lots of restaurants in Toronto we will try to detect locations that are not already crowded with restaurants. We are also particularly interested in areas with no Indian restaurants in vicinity. We would also prefer locations as close to city center as possible, assuming that first two conditions are met.

1.1 Background

There is a resturant contractor in one of the boroughs of Toronto (Scarborough). This contractor looking for the places such as Different types of Restaurants, Bakery, Breakfast Spot, Brewery and Café with fresh and high-quality. The contractor wants to build a restaurant for the Italian food lovers inside the borough, so that they will support more customers and also bring better "Quality of Service" to the there customers.

1.2 Problem Description

In this project we will try to find an location for a restaurant.

Since there are lots of restaurants in Toronto we will try to detect locations that are not already crowded with restaurants. We are also particularly interested in areas with no Indian restaurants

in vicinity. We would also prefer locations as close to city center as possible, assuming that first two conditions are met.

We will use our data science powers to generate a few most promissing neighborhoods based on this criteria. Advantages of each area will then be clearly expressed so that best possible final location can be chosen by stakeholders. this report will be targeted to stakeholders interested in opening an Italian restaurant in Toronto.

1.3 Target Audience

Target audience for this project will be all the stakeholders who want establish restaurants or café in Toronto neighborhood and want a less populated area so that the competition to other stakeholders will be less in comparison to others.

1.4 Success Criteria

The success criteria of this project will be a recommendation of area for new restaurant opening for the Italian food lovers on the basis of clusters in the nearby area.

2. DATA

2.1 Initial Dataset

number of existing restaurants in the neighborhood (any type of restaurant) number of and distance to Italian restaurants in the neighborhood, if any distance of neighborhood from city

center We decided to use regularly spaced grid of locations, centered around city center, to define our neighborhoods.

Following data sources will be needed to extract/generate the required information:

centers of candidate areas will be generated algorithmically and approximate addresses of centers of those areas will be obtained using Google Maps API reverse geocoding number of restaurants and their type and location in every neighborhood will be obtained using Foursquare API coordinate of Delhi center will be obtained using Google Maps API geocoding of well known toronto location.

Before we get the data and start exploring it, let's download all the dependencies that we will need.

```
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
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values
import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe
# 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
```

Downloading the data for segmentation

Use the Notebook to build the code to scrape the following Wikipedia page, https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M, in order to obtain the data that is in the table of postal codes and to transform the data into a pandas dataframe like the one shown below:

To create the dataframe: The dataframe will consist of three columns: PostalCode, Borough, and Neighborhood. Only process the cells that have an assigned borough. Ignore cells with a borough that is Not assigned. More than one neighborhood can exist in one postal code area. For example, in the table on the Wikipedia page, you will notice that M5A is listed twice and has two neighborhoods: Harbourfront and Regent Park. These two rows will be combined into one row with the neighborhoods separated with a comma as shown in row 11 in the above table. If a cell has a borough but a Not assigned neighborhood, then the neighborhood will be the same as the borough. So for the 9th cell in the table on the Wikipedia page, the value of the Borough and the Neighborhood columns will be Queen's Park. Clean your Notebook and add Markdown cells to explain your work and any assumptions you are making. In the last cell of your notebook, use the .shape method to print the number of rows of your dataframe. Note: There are different website scraping libraries and packages in Python. One of the most common packages is BeautifulSoup. Here is the package's main documentation page: http://beautiful-soup-4.readthedocs.io/en/latest/

#install Beautiful Soup

!pip install BeautifulSoup4

!pip install requests

GET DATA

now for downloading the data we will follow three steps first we will start with get html from

wikipedia then we will Use BeautifySoup to parse html data and then store parsed data into

Pandas DataFrame

1.get HTML from Wikipedia

```
from bs4 import BeautifulSoup
s1=requests.get("https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M")
soup = BeautifulSoup(s1.text, 'lxml')
```

2.use soup for parsing the data

```
d = []

col = []

table = soup.find(class_='wikitable')

for index, tr in enumerate(table.find_all('tr')):

    section = []

    for td in tr.find_all(['th','td']):

        section.append(td.text.rstrip())


    #First row of data is the header

    if (index == 0):

        col = section

    else:
```

d.append(section)

3.covert parse data to pandas dataframe

df = pd.DataFrame(data = d,columns = col)

df.head()

```
df = pd.DataFrame(data = d,columns = col)
df.head()
```

]:

	Postal Code	Borough	Neighbourhood
0	M1A	Not assigned	Not assigned
1	M2A	Not assigned	Not assigned
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Regent Park, Harbourfront

2.2 Data cleaning and Feature Engineering

More than one neighborhood can exist in one postal code area. For example, in the table on the Wikipedia page, you will notice that M5A is listed twice and has two neighborhoods: Harbourfront and Regent Park. These two rows will be combined into one row with the neighborhoods separated with a comma as shown in row 11 in the above table.

```
df1["Neighbourhood"] =df1.groupby("Postal Code")["Neighbourhood"].transform(lambda neigh: ', '.join(neigh))

#remove duplicates
df1= df1.drop_duplicates()


if(df1.index.name != 'Postal Code'):
    df1 = df1.set_index('Postal Code')

df1.head()
```

```
C:\Users\DELL PC\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
  """Entry point for launching an IPython kernel.
```

Postal Code	Borough	Neighbourhood
M3A	North York	Parkwoods
M4A	North York	Victoria Village
M5A	Downtown Toronto	Regent Park, Harbourfront
M6A	North York	Lawrence Manor, Lawrence Heights

If a cell has a borough but a Not assigned neighborhood, then the neighborhood will be the same as the borough.

```
df1['Neighbourhood'].replace("Not assigned", df1["Borough"],inplace=True)
df1.head()
```

Postal Code	Borough	Neighbourhood
M3A	North York	Parkwoods
M4A	North York	Victoria Village
M5A	Downtown Toronto	Regent Park, Harbourfront
M6A	North York	Lawrence Manor, Lawrence Heights
M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government

ne last cell of your notebook, use the .shape method to print the number of rows of your dataframe

```
df1.shape
```

```
(103, 2)
```

we need to have dataframe that contain both the data frames df1 and df_geo_coor so here it is

```
df_t = pd.merge(df1, df_geo_coor, how='inner', left_on = 'Postal Code', right_on = 'Postal Code')
```

```
df_t.shape
```

```
(103, 5)
```

```
df_t.head()
```

	Postal Code	Borough	Neighbourhood	Latitude	Longitude
0	M3A	North York	Parkwoods	43.753259	-79.329656
1	M4A	North York	Victoria Village	43.725882	-79.315572
2	M5A	Downtown Toronto	Regent Park, Harbourfront	43.654260	-79.360636
3	M6A	North York	Lawrence Manor, Lawrence Heights	43.718518	-79.464763
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494

```
df_t.to_excel("part_22.xlsx")
```

3.Methodology

The dataframe has 10 boroughs and 103 neighborhoods.

```
#count Borough and Neighborhood
df2.groupby('Borough').count()['Neighbourhood']
```

```
Borough
Central Toronto      9
Downtown Toronto    19
East Toronto         5
East York            3
Etobicoke           12
Mississauga          1
North York          24
Scarborough         17
West Toronto         6
York                 5
Name: Neighbourhood, dtype: int64
```

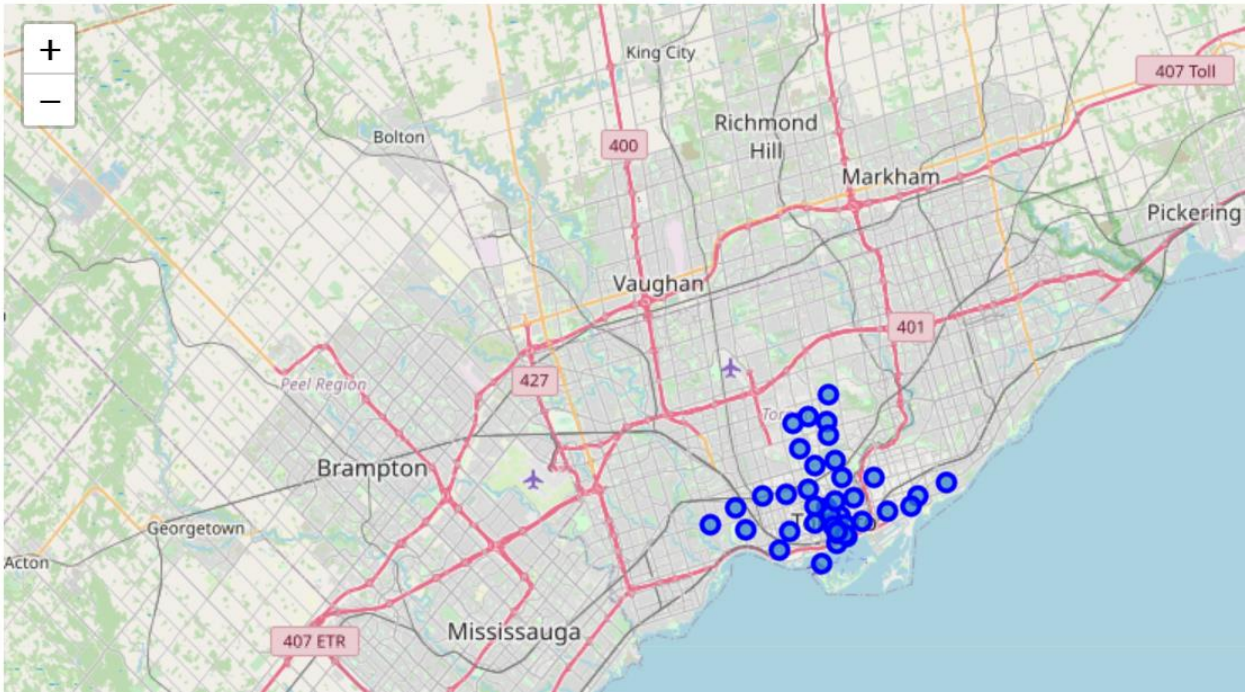
```
#Check the number of neighborhoods
print(df3.groupby('Borough').count()['Neighbourhood'])
```

```
Borough
Central Toronto      9
Downtown Toronto    19
East Toronto         5
West Toronto         6
Name: Neighbourhood, dtype: int64
```

now, as we have prepossessed data lets see all the neighbourhood in Toronto brought over the map. For ploting the map we will use folium. **Folium** is a powerful **Python** library that helps you create several types of Leaflet maps.To get an idea, just zoom/click around on the next map to

get an impression. The **Folium** github contains many other examples. By default, **Folium** creates a map in a separate HTML file.

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



Now with the help of foursquare api we will take the longitude and latitude of neighborhood venues in the radius of 500 and with the limit of 100. The **Foursquare** Places **API** provides location based experiences with diverse information about venues, users, photos, and check-ins. The **API** supports real time access to places, Snap-to-Place that assigns users to specific locations, and Geo-tag.

```
def getNearbyVenues(names, latitudes, longitudes, radius=500):

    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/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # 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'],
```

```
            v['venue']['categories'][0]['name']) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Neighbourhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

Now write the code to run the above function on each neighborhood and create a new dataframe called toronto_denc_venues

```
toronto_denc_venues.head()
```

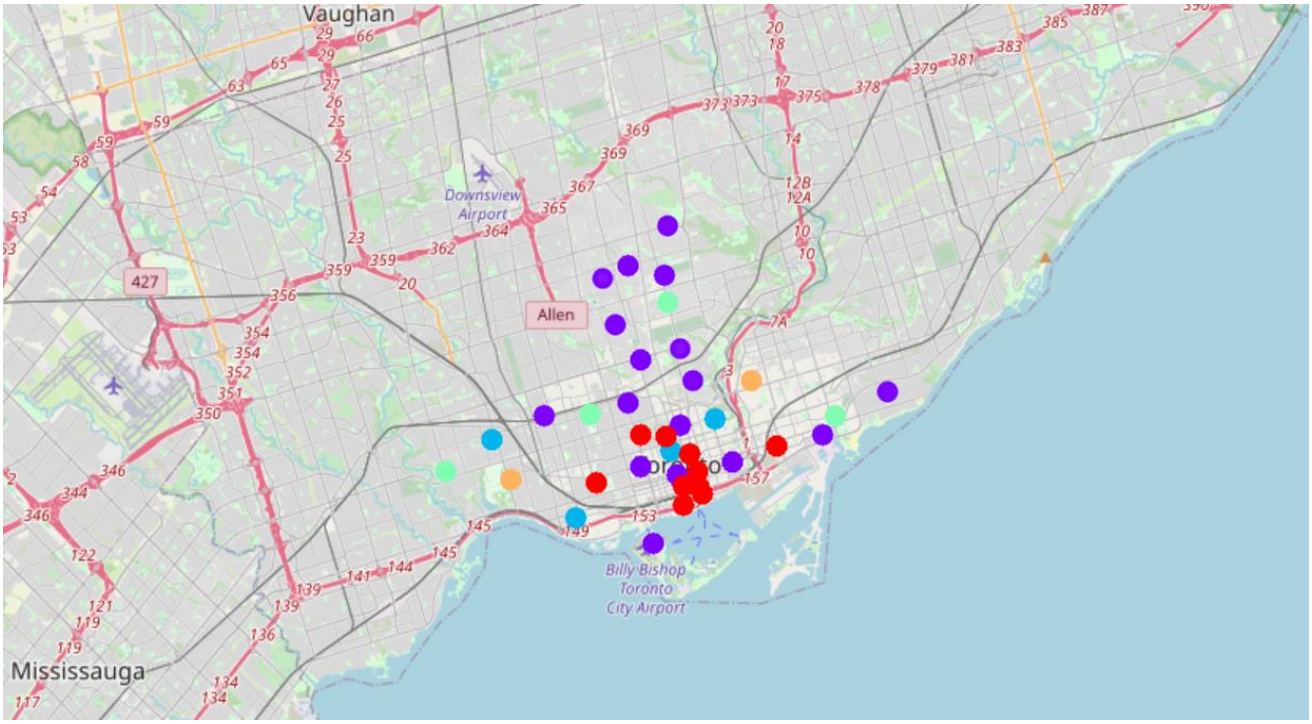
	Neighbourhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Regent Park, Harbourfront	43.65426	-79.360636	Roselle Desserts	43.653447	-79.362017	Bakery
1	Regent Park, Harbourfront	43.65426	-79.360636	Tandem Coffee	43.653559	-79.361809	Coffee Shop
2	Regent Park, Harbourfront	43.65426	-79.360636	Cooper Koo Family YMCA	43.653249	-79.358008	Distribution Center
3	Regent Park, Harbourfront	43.65426	-79.360636	Body Blitz Spa East	43.654735	-79.359874	Spa
4	Regent Park, Harbourfront	43.65426	-79.360636	Impact Kitchen	43.656369	-79.356980	Restaurant

Now we will check if the results contain "Italian Restaurants" for that we will use following command.

"Italian Restaurant" in toronto_denc_venues['Venue Category'].unique()

4.Analysis

Now, we focus on the centers of clusters and compare them for their "Total Restaurants" and their "Total Joints". The group which its center has the highest "Total Sum" will be our best recommendation to the contractor. {Note: Total Sum = Total Restaurants + Total Joints.} This algorithm although is pretty straightforward yet is strongly powerful. Run k-means to cluster the neighborhoods in Toronto into 5 clusters and then we will plot the map with the five different colour clusters, here this dark blue cluster is cluster 0 ,orange cluster is cluster 1 then red colour cluster is cluster 2, sky blue colour cluster is cluster 3 and light green cluster is cluster 4.



5. Observation

Most of Italian restaurants are in Cluster 0 and lowest in Cluster 4 and cluster 2 areas. Also, there are good opportunities to open restrurant in parkside drive and bayview avenue bloor street areas as the competition seems to be low. Looking at nearby venues, it seems Cluster 4 might be a good location as there are not a lot of italian restaurants in these areas. Therefore, this project recommends the entrepreneur to open an authentic Burmese restaurant in these locations with little to no competition. Nonetheless, if the food is authentic, affordable and good taste, I am confident that it will have great following everywhere.