Battle of Neighbourhoods (Week 1)

May 13, 2020

1 Battle of Neighbourhoods (Week 1)

2 1. Description of the Problem and Discussion of the Background

2.1 A. Introduction

2.2 Prospects of Opening an Indian Restaurant in the City of Toronto, Canada.

According to a study, Canada contains the worlds's eighth largest Indian diaspora. There has been a 74% rise in the Indian Immigrants since 2001.Not only that, Indians account for one-fourth of Immigrants Population in Canada. Canada is also the third most popular country to pursue Higher studies for Indian students. Toronto is the financial capital of Canada and you can find Indians working in every sector making it one of the best places to set up an Indian Restaurant.

We will work trough this project step by step. For this week I shall describe the initial data preparation and future steps for the project.

1.Obtain the Data

1.a. Name of the Boroughs and Neighbourhoods from web scrapping

1.b. Obtain information about selected Boroughs in Toronto.

1.c. Use Foresquare Data to obtain info about restaurants.

2.Data Visualization and Some Simple Statistical Analysis.

3.Analysis Using Clustering, Specially K-Means Clustering.

3.a. Maximize the number of clusters.

3.b. Visualization using Chloropleth Map

4.Compare the Neighborhoods to Find the Best Place for Starting up a Restaurant.

5.Inference From these Results and related Conclusions.

Target Audience

1.Business personnel who wants to invest in or open a restaurant.
2.This analysis will be a comprehensive guide to start or expand restaurants targeting the large

pool of office workers in Toronto.
3.New graduates, to find reasonable lunch/breakfast place close to office.

This analysis will give an idea, how benificial it is to open a restaurant and what are the pros and cons of this business.

2.3 2. Initial Data Preparation (Week 1)

2.1. Get The Names of Boroughs, Neighbourhoods and Land Market Price from Wikipedia.

2.2. Processing the Information From Wiki to select areas of interest.

2.3. Get the Coordinates of the Boroughs.

I have described these steps detailed as a part of Week 1 Assignment in a python notebook, and also made a more comprehensible .pdf file.

So here we start with using Foursquare and use many Exploratory Data Analysis Techniques to learn from data and visualize to strengthen our understanding.

2.3.1 Importing Libraries for data analysis.

[1]:
```
import pandas as pd # library for data analsysis
import numpy as np
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
#!conda install -c conda-forge geopy --yes # uncomment this line if you haven't␣
 ↪completed the Foursquare API lab
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
!pip install folium
```

```
Requirement already satisfied: folium in c:\users\nidhi
shetty\anaconda3\lib\site-packages (0.11.0)
Requirement already satisfied: branca>=0.3.0 in c:\users\nidhi
shetty\anaconda3\lib\site-packages (from folium) (0.4.1)
Requirement already satisfied: jinja2>=2.9 in c:\users\nidhi
shetty\anaconda3\lib\site-packages (from folium) (2.11.1)
Requirement already satisfied: requests in c:\users\nidhi
shetty\anaconda3\lib\site-packages (from folium) (2.22.0)
```

```
Requirement already satisfied: numpy in c:\users\nidhi
shetty\anaconda3\lib\site-packages (from folium) (1.18.1)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\nidhi
shetty\anaconda3\lib\site-packages (from jinja2>=2.9->folium) (1.1.1)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\nidhi
shetty\anaconda3\lib\site-packages (from requests->folium) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\nidhi
shetty\anaconda3\lib\site-packages (from requests->folium) (2019.11.28)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\users\nidhi shetty\anaconda3\lib\site-packages (from requests->folium)
(1.25.8)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\nidhi
shetty\anaconda3\lib\site-packages (from requests->folium) (2.8)
```

2.3.2 2.1. Get The Names of Boroughs, Neighbourhoods and Land Market Price from Wikipedia.

```
[2]: tpc1 = pd.read_csv(r'C:\Users\Nidhi Shetty\Desktop\TorontoPC.csv')
     tpc1.head()
```

```
[2]:   Postal Code       Borough  \
     0         M9Z  Not assigned
     1         M9Y  Not assigned
     2         M9X  Not assigned
     3         M9W     Etobicoke
     4         M9V     Etobicoke

                                            Neighborhood  Market_Value_per Sq.ft
     0                                               NaN                     NaN
     1                                               NaN                     NaN
     2                                               NaN                     NaN
     3                                         Northwest                     NaN
     4  South Steeles / Silverstone / Humbergate / Jam...                     NaN
```

```
[3]: tpc2= pd.read_csv(r'C:\Users\Nidhi Shetty\Desktop\Geospatial_Coordinates.csv')
     tpc2.head()
```

```
[3]:   Postal Code   Latitude  Longitude
     0         M1B  43.806686 -79.194353
     1         M1C  43.784535 -79.160497
     2         M1E  43.763573 -79.188711
     3         M1G  43.770992 -79.216917
     4         M1H  43.773136 -79.239476
```

```
[4]: #Replacing the "Not assigned" values in Column 'Borough' to the standard NaN.
     tpc1["Borough"].replace("Not assigned", np.nan, inplace = True)
```

```
#Dropping the rows with NaN values in the 'Borough'.
tpc1.dropna(subset=["Borough"], axis=0, inplace=True)

# reset index, because we dropped rows
tpc1.reset_index(drop=True, inplace=True)
```

[5]:

```
#Replacing the '/' by ',' .
tpc1['Neighborhood']=tpc1['Neighborhood'].str.replace('/', ',')
tpc1.head()
```

[5]:

```
  Postal Code    Borough                                        Neighborhood  \
0         M9W  Etobicoke                                           Northwest   
1         M9V  Etobicoke  South Steeles , Silverstone , Humbergate , Jam...   
2         M9R  Etobicoke  Kingsview Village , St. Phillips , Martin Grov...   
3         M9P  Etobicoke                                           Westmount   
4         M9N       York                                              Weston   

   Market_Value_per Sq.ft  
0                     NaN  
1                     NaN  
2                     NaN  
3                     NaN  
4                     NaN  
```

[6]:

```
#Merging the two dataframes using 'inner' join operator
tpcfinal= tpc1.merge(tpc2, how ='inner')
tpcfinal.head()
tpcfinal.rename(columns={'Neighborhood': 'Neighbourhood'}, inplace=True)
tpcfinal.head()
```

[6]:

```
  Postal Code    Borough                                       Neighbourhood  \
0         M9W  Etobicoke                                           Northwest   
1         M9V  Etobicoke  South Steeles , Silverstone , Humbergate , Jam...   
2         M9R  Etobicoke  Kingsview Village , St. Phillips , Martin Grov...   
3         M9P  Etobicoke                                           Westmount   
4         M9N       York                                              Weston   

   Market_Value_per Sq.ft   Latitude  Longitude  
0                     NaN  43.706748 -79.594054  
1                     NaN  43.739416 -79.588437  
2                     NaN  43.688905 -79.554724  
3                     NaN  43.696319 -79.532242  
4                     NaN  43.706876 -79.518188  
```

[7]:

```
print('The dataframe has {} boroughs and {} neighborhoods.'.format(
        len(tpcfinal['Borough'].unique()),
        tpcfinal.shape[0]
    )
)
```

```
The dataframe has 10 boroughs and 103 neighborhoods.
```

2.3.3 2.2. Processing the Information From Wiki to select areas of interest.

[8]:
```
#Next job is to select boroughs containing the word Toronto

# check the borough names at first
unique_boroughs = list(tpcfinal.Borough.unique())
print(unique_boroughs)
# select boroughs with word Toronto in it
Tor_boroughs = ['East Toronto', 'Central Toronto', 'Downtown Toronto', 'West␣
 ↪Toronto']
```

```
['Etobicoke', 'York', 'North York', 'East Toronto', 'Mississauga', 'Downtown
Toronto', 'West Toronto', 'Central Toronto', 'East York', 'Scarborough']
```

2.3.4 2.3. Get the Coordinates of Toronto City.

[9]:
```
address = 'Toronto City, CA'

geolocator = Nominatim(user_agent="TO_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geographical coordinate Toronto City are {}, {}.'.format(latitude,␣
 ↪longitude))
```

```
The geographical coordinate Toronto City are 43.6534817, -79.3839347.
```

2.3.5 2.4 Creating a data frame and a map for Toronto.

[10]:
```
Tor_series = tpcfinal["Borough"].isin(Tor_boroughs)
Toronto = tpcfinal[Tor_series]



Toronto.index = np.arange(0, len(Toronto))
print ("Shape of Dataframe with boroughs containing the word Toronto: ", Toronto.
 ↪shape)
Toronto.head()
```

```
Shape of Dataframe with boroughs containing the word Toronto:  (39, 6)
```

[10]:

```
  Postal Code           Borough                                 Neighbourhood  \
0         M7Y      East Toronto         Business reply mail Processing Centre
1         M7A  Downtown Toronto  Queen's Park , Ontario Provincial Government
2         M6S      West Toronto                           Runnymede , Swansea
3         M6R      West Toronto                       Parkdale , Roncesvalles
4         M6P      West Toronto                High Park , The Junction South

   Market_Value_per Sq.ft   Latitude  Longitude
0                    90.0  43.662744 -79.321558
1                   200.0  43.662301 -79.389494
2                   100.0  43.651571 -79.484450
3                   100.0  43.648960 -79.456325
4                   100.0  43.661608 -79.464763
```

[12]:

```
import folium
# create map of Toronto using latitude and longitude values
map_Toronto = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, borough, neighborhood in zip(Toronto['Latitude'],␣
 ↪Toronto['Longitude'], Toronto['Borough'], Toronto['Neighbourhood']):
    label = '{}, {}'.format(neighborhood, borough)
    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)

map_Toronto
```

[12]:

```
<folium.folium.Map at 0x24eac428208>
```

[ ]:

[ ]:

[ ]: