CAPSTONE PROJECT : BATTLE OF THE NEIGHBOURHOODS

IDENTIFYING BEST NEIGHBORHOOD TO START A NEW RESTAUARNT

1. Introduction/Business Problem:

The economic recession that put America on its knees hardly touched Toronto. Steadied by strong, highly regulated banks and buoyed by an educated workforce, Canada's largest city is open for business.

Toronto regularly tops most livable cities lists, in part because it has a large percentage of parkland. The immigrant population is so large that the city has several Chinatowns and Little Italys, in addition to dozens of ethnic neighborhoods.

The idea of the project is to help the stake holders by identifying the appropriate location to start his/her new restaurant. From the above points identifying the no. of neighborhoods, population and income plays an important role in establishing a new restaurant

Once the neighborhoods have been listed, we need to find out in which neighborhoods there is less competition. It means that we have to find out what neighborhoods contain the lowest number of existing restaurants of the same type as the one we want to open using FoursquareAPI explore query.

2. Data Set Description

The neighbourhood profiles were developed to help government and community agencies with their local planning, by providing socio-economic data at a meaningful geographic area. The boundaries of these social planning neighbourhoods do not change over time, allowing researchers to perform longitudinal studies see the changes in each area.

The list of neighbourhoods, and the demographic data associated to each neighbourhood, has been made available by the city of Toronto here :

- https://www.toronto.ca/ext/open_data/catalog/data_set_files/2016_neighbourhood_profiles.csv

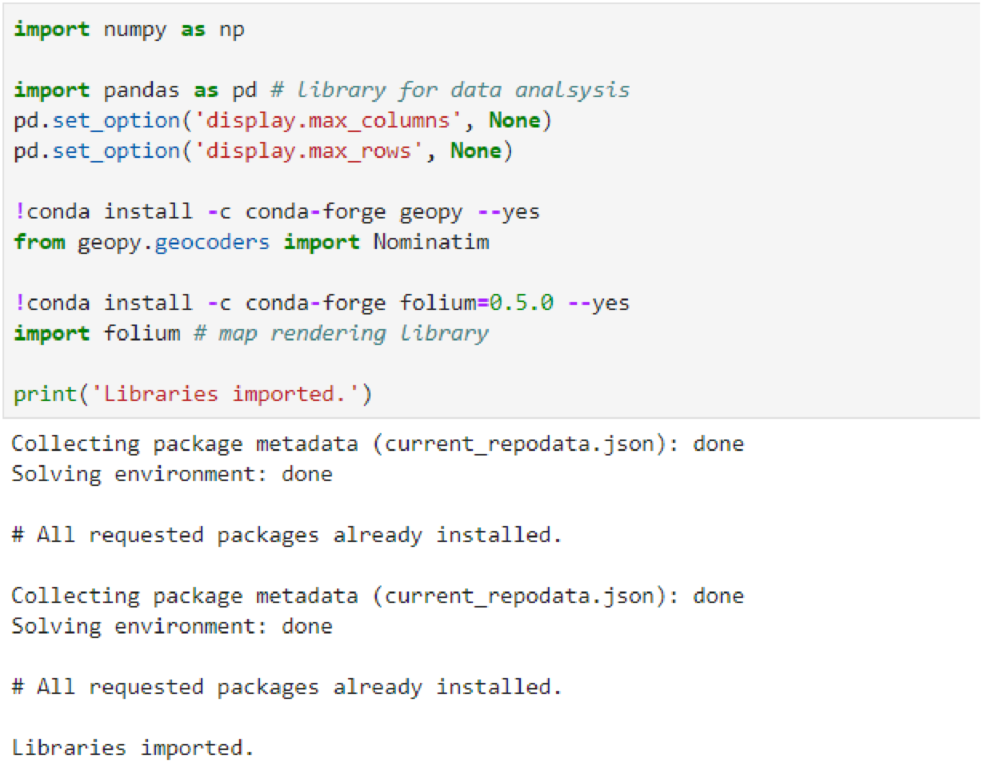
The Toronto demographic dataset contains multiple features such as :

- Citizenship
- **Income**
- Languages / Mother tongue
- Marital status
- **Neighbourhood information**
- **Population**
- Etc

We query FoursquareAPI suplying the neighbourhood's information (coordinates calculated with the **Geocoder** package), the radius of scan, and the limit of number of venues we want to retrieve.

3. Methodology

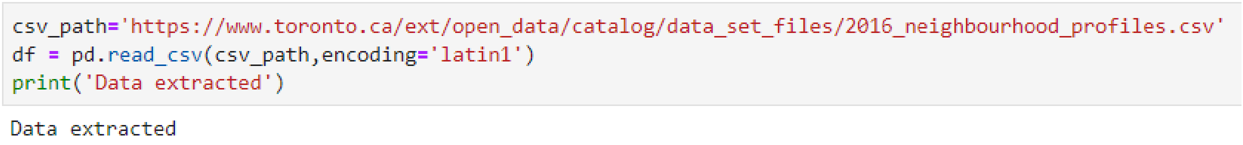
1. All the necessary packages required are loaded as shown below



```
import numpy as np

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

!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.')
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

Libraries imported.
```

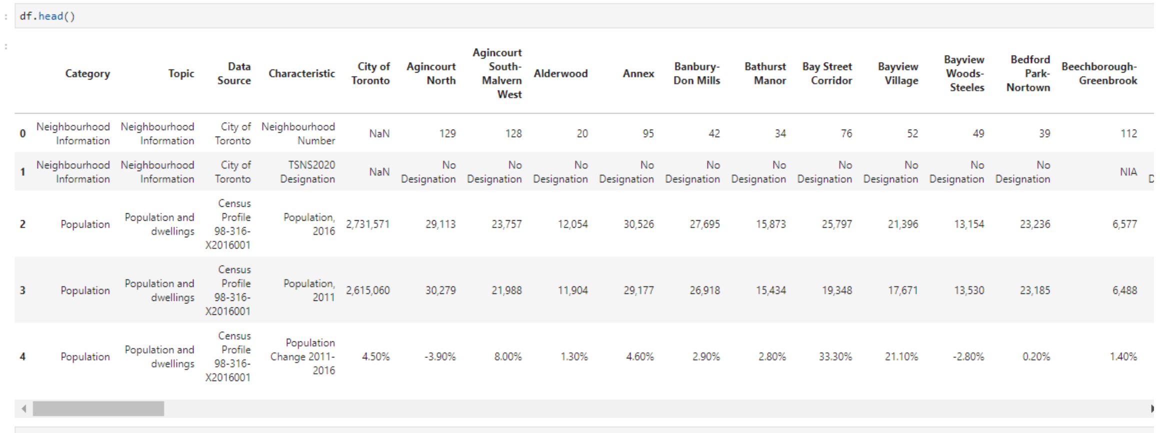
2. We extract the data from the path given as shown below



```
csv_path='https://www.toronto.ca/ext/open_data/catalog/data_set_files/2016_neighbourhood_profiles.csv'
df = pd.read_csv(csv_path,encoding='latin1')
print('Data extracted')
```

```
Data extracted
```

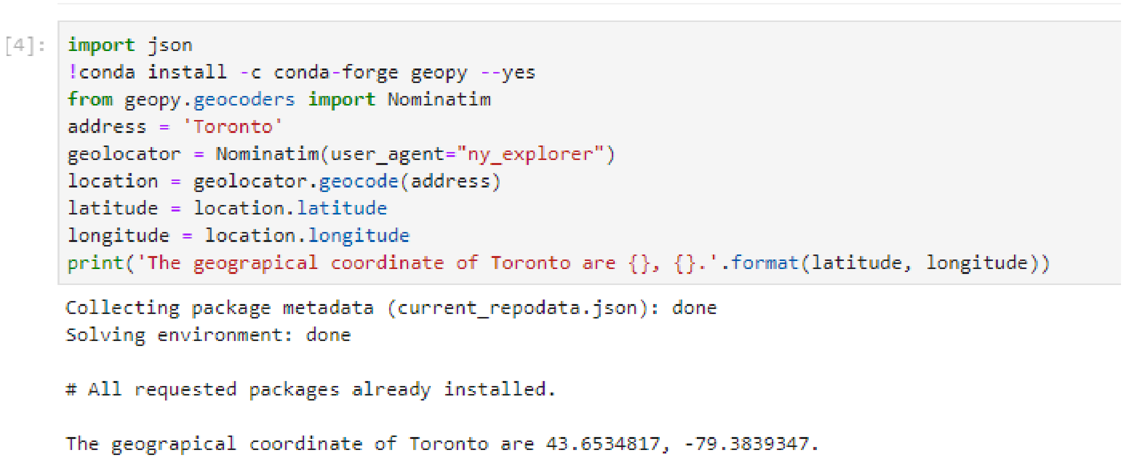
3. An instance of the data can be seen by using head() function as shown below



```
df.head()
```

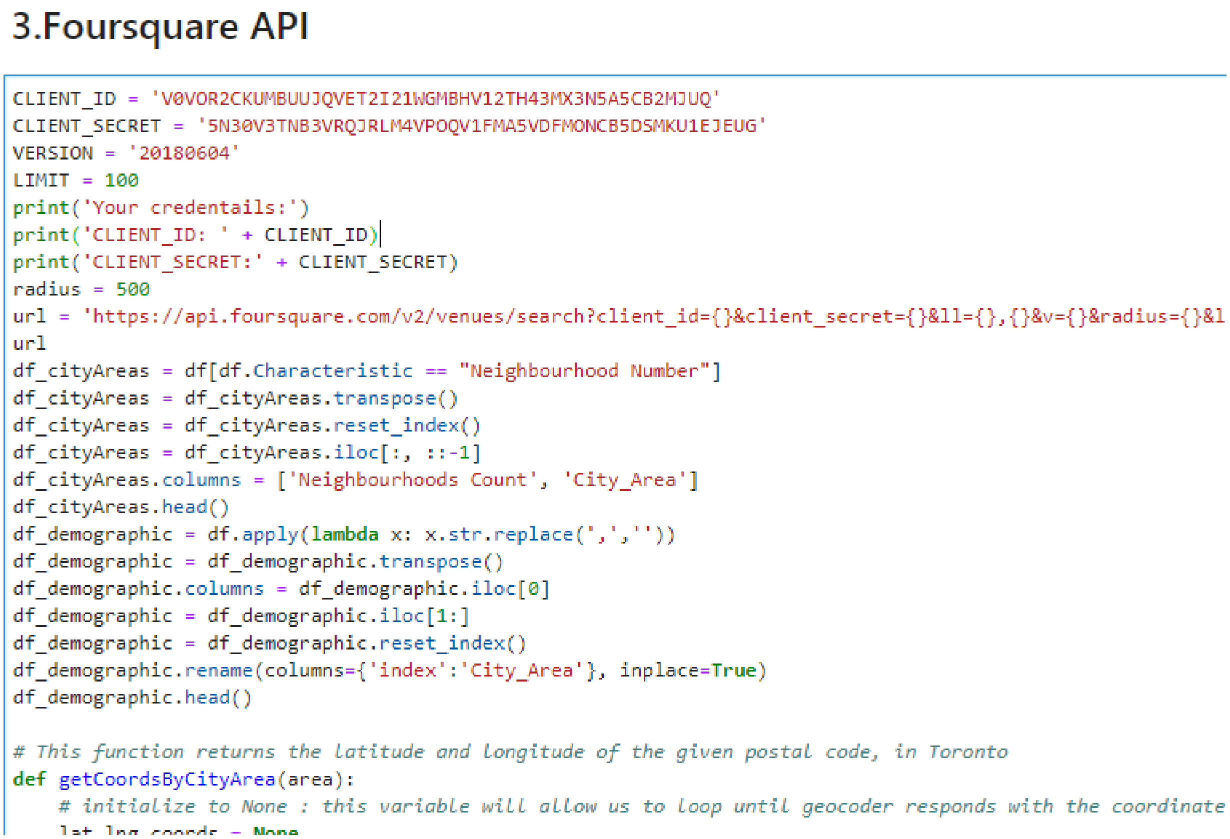
	Category	Topic	Data Source	Characteristic	City of Toronto	Agincourt North	Agincourt South-Malvern West	Alderwood	Annex	Banbury-Don Mills	Bathurst Manor	Bay Street Corridor	Bayview Village	Bayview Woods-Steeles	Bedford Park-Nortown	Beechborough-Greenbrook
0	Neighbourhood Information	Neighbourhood Information	City of Toronto	Neighbourhood Number	NaN	129	128	20	95	42	34	76	52	49	39	112
1	Neighbourhood Information	Neighbourhood Information	City of Toronto	TSNS2020 Designation	NaN	No Designation	No Designation	No Designation	No Designation	No Designation	No Designation	No Designation	No Designation	No Designation	No Designation	NIA
2	Population	Population and dwellings	Census Profile 98-316-X2016001	Population, 2016	2,731,571	29,113	23,757	12,054	30,526	27,695	15,873	25,797	21,396	13,154	23,236	6,577
3	Population	Population and dwellings	Census Profile 98-316-X2016001	Population, 2011	2,615,060	30,279	21,988	11,904	29,177	26,918	15,434	19,348	17,671	13,530	23,185	6,488
4	Population	Population and dwellings	Census Profile 98-316-X2016001	Population Change 2011-2016	4.50%	-3.90%	8.00%	1.30%	4.60%	2.90%	2.80%	33.30%	21.10%	-2.80%	0.20%	1.40%

4. The coordinates of Toronto i.e., the geographical locatiojn is displayed as shown below



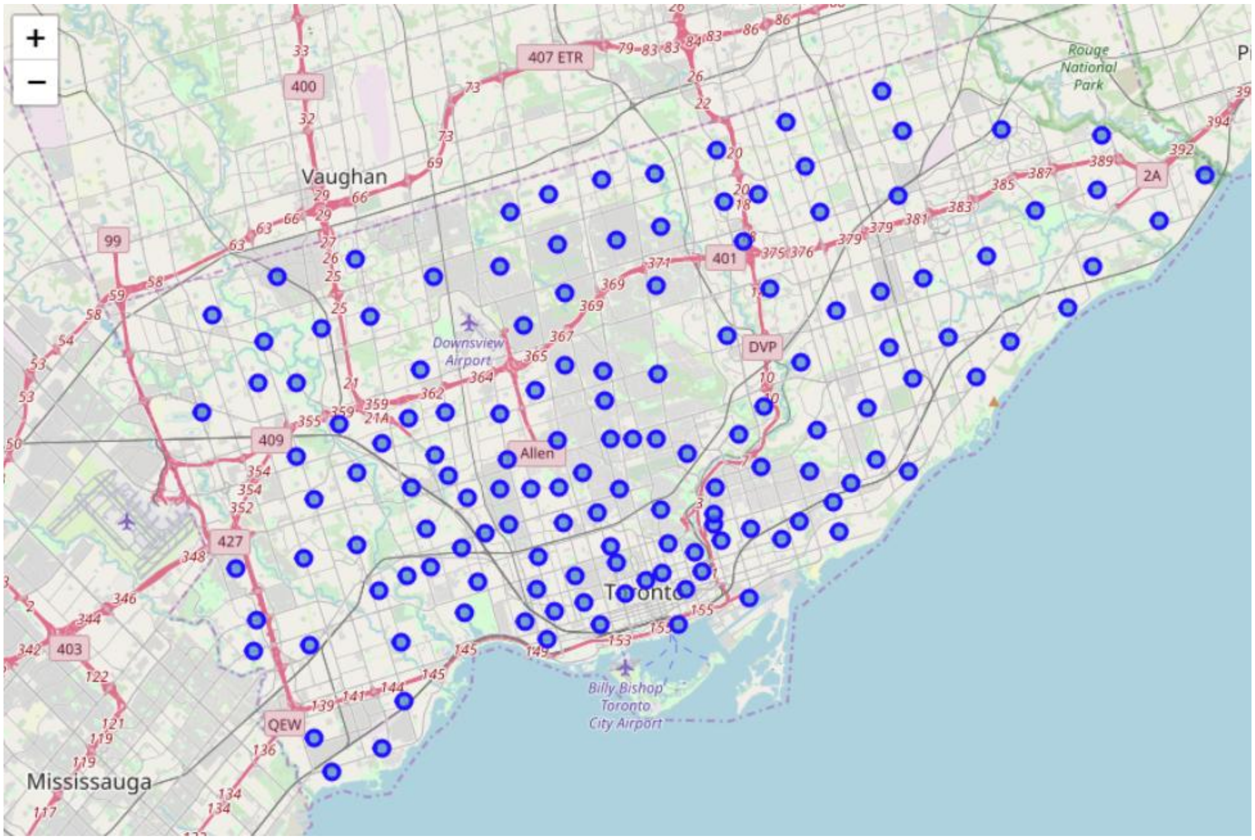
```
[4]: import json
     !conda install -c conda-forge geopy --yes
     from geopy.geocoders import Nominatim
     address = 'Toronto'
     geolocator = Nominatim(user_agent="ny_explorer")
     location = geolocator.geocode(address)
     latitude = location.latitude
     longitude = location.longitude
     print('The geograpical coordinate of Toronto are {}, {}.'.format(latitude, longitude))

     Collecting package metadata (current_repodata.json): done
     Solving environment: done

     # All requested packages already installed.

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

5. Using Foursquare API we the neighbor hoods and the same is done , some part of the code is shown below and the neighbhorhoods are shown in the map

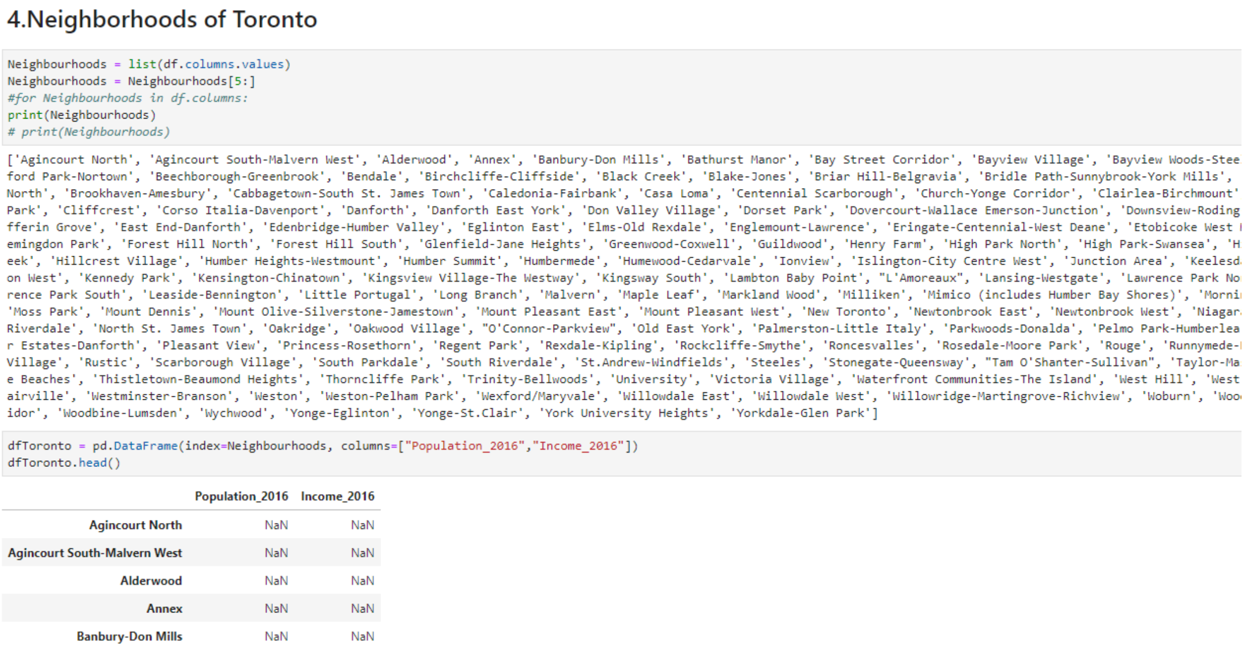


3.Foursquare API

```
CLIENT_ID = 'V0VOR2CKUMBUUJQVET2I21WGMBHV12TH43MX3N5A5CB2MJUQ'
CLIENT_SECRET = '5N30V3TNB3VRQJRLM4VPOQV1FMA5VDFMONCB5DSMKU1EJEUG'
VERSION = '20180604'
LIMIT = 100
print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
radius = 500
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&v={}&radius={}&l
url
df_cityAreas = df[df.Characteristic == "Neighbourhood Number"]
df_cityAreas = df_cityAreas.transpose()
df_cityAreas = df_cityAreas.reset_index()
df_cityAreas = df_cityAreas.iloc[:, ::-1]
df_cityAreas.columns = ['Neighbourhoods Count', 'City_Area']
df_cityAreas.head()
df_demographic = df.apply(lambda x: x.str.replace(',',''))
df_demographic = df_demographic.transpose()
df_demographic.columns = df_demographic.iloc[0]
df_demographic = df_demographic.iloc[1:]
df_demographic = df_demographic.reset_index()
df_demographic.rename(columns={'index':'City_Area'}, inplace=True)
df_demographic.head()

# This function returns the latitude and longitude of the given postal code, in Toronto
def getCoordsByCityArea(area):
    # initialize to None : this variable will allow us to loop until geocoder responds with the coordinate
    lat_lng_coords = None
```



6. All the neighborhoods in Toranto are dispalyed



4.Neighborhoods of Toronto

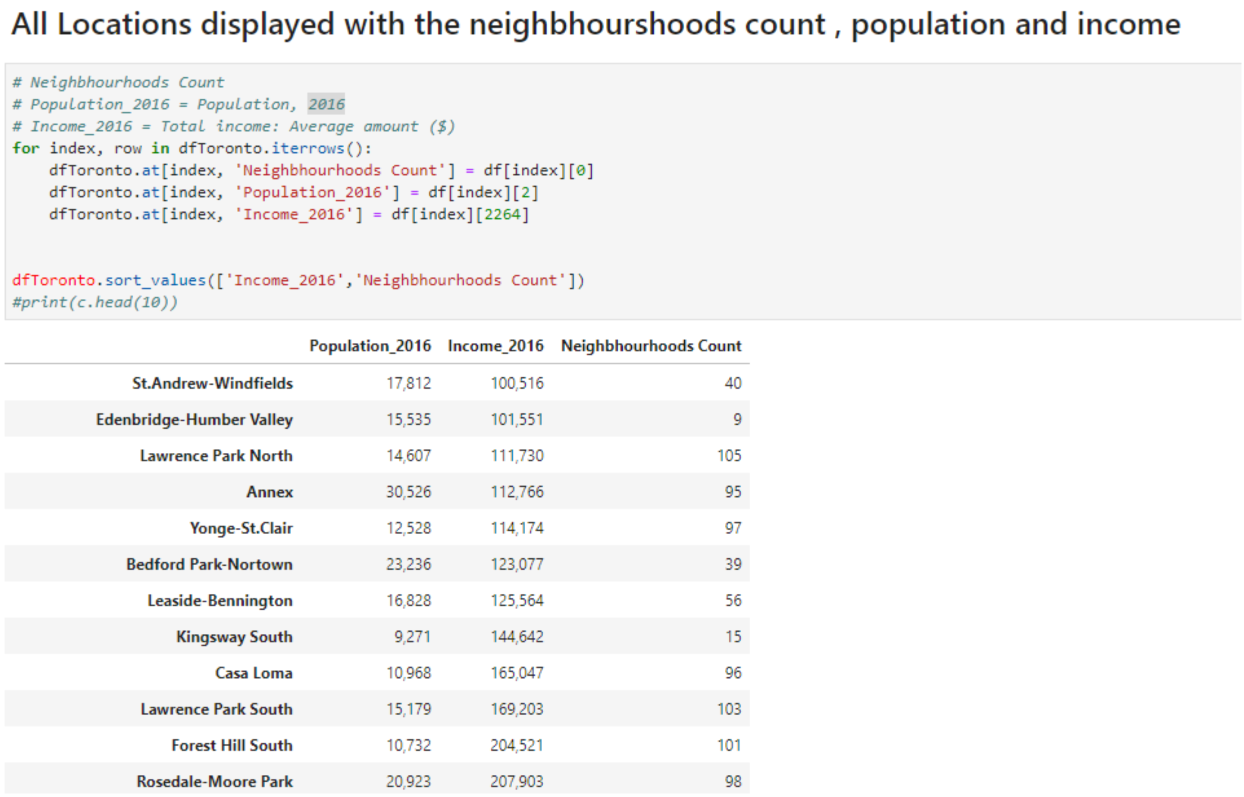
```
Neighbourhoods = list(df.columns.values)
Neighbourhoods = Neighbourhoods[5:]
#for Neighbourhoods in df.columns:
print(Neighbourhoods)
# print(Neighbourhoods)
```

```
['Agincourt North', 'Agincourt South-Malvern West', 'Alderwood', 'Annex', 'Banbury-Don Mills', 'Bathurst Manor', 'Bay Street Corridor', 'Bayview Village', 'Bayview Woods-Stee
ford Park-Nortown', 'Beechborough-Greenbrook', 'Bendale', 'Birchcliffe-Cliffside', 'Black Creek', 'Blake-Jones', 'Briar Hill-Belgravia', 'Bridle Path-Sunnybrook-York Mills',
North', 'Brookhaven-Amesbury', 'Cabbagetown-South St. James Town', 'Caledonia-Fairbank', 'Casa Loma', 'Centennial Scarborough', 'Church-Yonge Corridor', 'Clairlea-Birchmount'
Park', 'Cliffcrest', 'Corso Italia-Davenport', 'Danforth', 'Danforth East York', 'Don Valley Village', 'Dorset Park', 'Dovercourt-Wallace Emerson-Junction', 'Downsview-Roding
fferin Grove', 'East End-Danforth', 'Edenbridge-Humber Valley', 'Eglinton East', 'Elms-Old Rexdale', 'Englemount-Lawrence', 'Eringate-Centennial-West Deane', 'Etobicoke West 
emingdon Park', 'Forest Hill North', 'Forest Hill South', 'Glenfield-Jane Heights', 'Greenwood-Coxwell', 'Guildwood', 'Henry Farm', 'High Park North', 'High Park-Swansea', 'H
eek', 'Hillcrest Village', 'Humber Heights-Westmount', 'Humber Summit', 'Humbermede', 'Humewood-Cedarvale', 'Ionview', 'Islington-City Centre West', 'Junction Area', 'Keelesd
on West', 'Kennedy Park', 'Kensington-Chinatown', 'Kingsview Village-The Westway', 'Kingsway South', 'Lambton Baby Point', "L'Amoreaux", 'Lansing-Westgate', 'Lawrence Park No
rence Park South', 'Leaside-Bennington', 'Little Portugal', 'Long Branch', 'Malvern', 'Maple Leaf', 'Markland Wood', 'Milliken', 'Mimico (includes Humber Bay Shores)', 'Morni
'Moss Park', 'Mount Dennis', 'Mount Olive-Silverstone-Jamestown', 'Mount Pleasant East', 'Mount Pleasant West', 'New Toronto', 'Newtonbrook East', 'Newtonbrook West', 'Niagar
Riverdale', 'North St. James Town', 'Oakridge', 'Oakwood Village', "O'Connor-Parkview", 'Old East York', 'Palmerston-Little Italy', 'Parkwoods-Donalda', 'Pelmo Park-Humberlea
r Estates-Danforth', 'Pleasant View', 'Princess-Rosethorn', 'Regent Park', 'Rexdale-Kipling', 'Rockcliffe-Smythe', 'Roncesvalles', 'Rosedale-Moore Park', 'Rouge', 'Runnymede-
Village', 'Rustic', 'Scarborough Village', 'South Parkdale', 'South Riverdale', 'St.Andrew-Windfields', 'Steeles', 'Stonegate-Queensway', "Tam O'Shanter-Sullivan", 'Taylor-Ma
e Beaches', 'Thistletown-Beaumond Heights', 'Thorncliffe Park', 'Trinity-Bellwoods', 'University', 'Victoria Village', 'Waterfront Communities-The Island', 'West Hill', 'West
airville', 'Westminster-Branson', 'Weston', 'Weston-Pelham Park', 'Wexford/Maryvale', 'Willowdale East', 'Willowdale West', 'Willowridge-Martingrove-Richview', 'Woburn', 'Woo
idor', 'Woodbine-Lumsden', 'Wychwood', 'Yonge-Eglinton', 'Yonge-St.Clair', 'York University Heights', 'Yorkdale-Glen Park']
```

```
dfToronto = pd.DataFrame(index=Neighbourhoods, columns=["Population_2016","Income_2016"])
dfToronto.head()
```

	Population_2016	Income_2016
Agincourt North	NaN	NaN
Agincourt South-Malvern West	NaN	NaN
Alderwood	NaN	NaN
Annex	NaN	NaN
Banbury-Don Mills	NaN	NaN

7. The neighborhoods are now displayed with their income and population in the sorting order of income and then neighborhood count

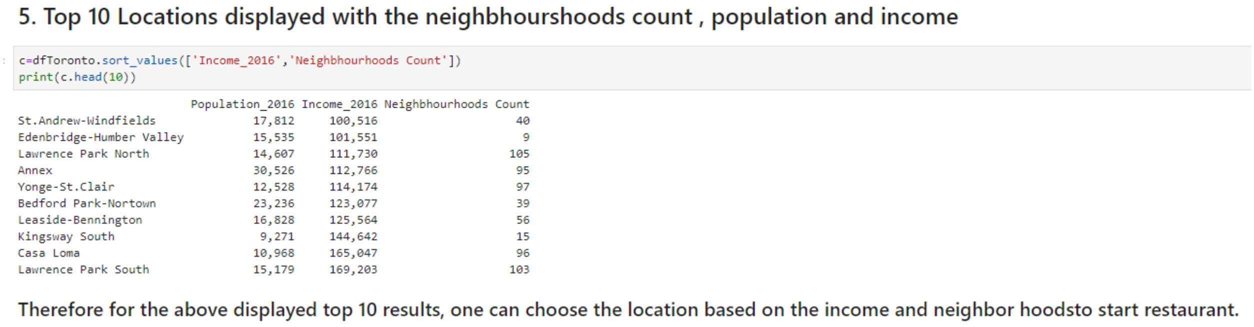


All Locations displayed with the neighbhourshoods count , population and income

```
# Neighbhourhoods Count
# Population_2016 = Population, 2016
# Income_2016 = Total income: Average amount ($)
for index, row in dfToronto.iterrows():
    dfToronto.at[index, 'Neighbhourhoods Count'] = df[index][0]
    dfToronto.at[index, 'Population_2016'] = df[index][2]
    dfToronto.at[index, 'Income_2016'] = df[index][2264]


dfToronto.sort_values(['Income_2016','Neighbhourhoods Count'])
#print(c.head(10))
```

	Population_2016	Income_2016	Neighbhourhoods Count
St.Andrew-Windfields	17,812	100,516	40
Edenbridge-Humber Valley	15,535	101,551	9
Lawrence Park North	14,607	111,730	105
Annex	30,526	112,766	95
Yonge-St.Clair	12,528	114,174	97
Bedford Park-Nortown	23,236	123,077	39
Leaside-Bennington	16,828	125,564	56
Kingsway South	9,271	144,642	15
Casa Loma	10,968	165,047	96
Lawrence Park South	15,179	169,203	103
Forest Hill South	10,732	204,521	101
Rosedale-Moore Park	20,923	207,903	98

8. The top 10 neighborhoods



5. Top 10 Locations displayed with the neighbhourshoods count , population and income

```
c=dfToronto.sort_values(['Income_2016','Neighbhourhoods Count'])
print(c.head(10))
```

```
                         Population_2016 Income_2016 Neighbhourhoods Count
St.Andrew-Windfields              17,812     100,516                    40
Edenbridge-Humber Valley          15,535     101,551                     9
Lawrence Park North               14,607     111,730                   105
Annex                             30,526     112,766                    95
Yonge-St.Clair                    12,528     114,174                    97
Bedford Park-Nortown              23,236     123,077                    39
Leaside-Bennington                16,828     125,564                    56
Kingsway South                     9,271     144,642                    15
Casa Loma                         10,968     165,047                    96
Lawrence Park South               15,179     169,203                   103
```

Therefore for the above displayed top 10 results, one can choose the location based on the income and neighbor hoodsto start restaurant.

5.Conclusion

In this project, I tried to find the neighbourhoods based on their population and income. This gives the insight of various neighbourhoods for the opening of a new restaurant.

Then, I identified top 10 neighbourhoods with the adequate information such as income and the count, within these adequate neighbourhoods, one can gain profits if a restaurant is opened.

Catering services can use similar data analysis in order to find the best spots to open a retail shop.