Best Prospective Business to Start Out in Toronto

Jeffrey Chai Zhi Yang
February 9, 2021

1. Introduction

1.1 Background

As an entrepreneur, John wishes to start his own business in Toronto, but he needs ideas on what is the most popular business to do in the city at the moment. He has been doing his own primary market research around the neighbourhood but has deemed it as a very time inefficient method of doing so. Then, he thought of perhaps using a data driven approach to assist him in the matter.

1.2 Business Problem

My friend is looking for recommendation to do a business in Toronto and would like to find out what is the most established commerciality in a given district of Toronto. This is based on his presumption that the most common venues equates to the most popular business that he could start on.

1.3 Motivation

The project aims to find out what is the most visited venues in the majority of the neighbourhoods (namely postal code M) in Toronto so that this can give John some form of confidence, driven by data analytics, that he could probably set up a similar business given the highest level of demand for that given establishment. For instance, assume that for most of the neighbourhoods have gym as its most common venue, it could imply that there is greatest amount of demand for such an amenity and that it could be the most prospective business he could give a try on.

2. Data

2.1 Data Sources

There are 3 data sources required for the analysis:

- Source 1: List of Postal Codes (only M) of Canada – from a Wikipedia page. Since the data is from a website and it is in a tabular format, it requires some form of data scraping process and then followed by transformation of the scraped data into dataframes.
- Source 2: Geographical coordinates of each postal code (only M) in Toronto – offline downloaded in csv format.
- Source 3: Information on each Neighbourhood in Toronto - Data to be obtained through API call.

2.2 Data Cleaning and Transformation Processes

As mentioned earlier, There are 3 data sources required and each of them has the following cleaning processes performed on prior to the actual analysis of the data:

<u>Source 1: List of Postal Codes (only M) of Canada</u>

The data is scraped using pandas's 'read_html' function. The data is in tabular format and is consisting of 3 data fields – *Postal Code, Borough, Neighbourhood*.

```
In [2]: # website scraping
url = 'https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M'
df = pd.read_html(url)
df = df[0]
df.head()
```

Out[2]:

	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

After which, the rows where Borough is not assigned are dropped and not used for analysis later on.

```
In [3]: # Drop borough cells with note assigned
df = df.drop(df[df.Borough == 'Not assigned'].index)
df.head()
```

Out[3]:

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

<u>Source 2: Geographical coordinates of each postal code (only M) in Toronto</u>

As this is in CSV format, it can be read and transformed into a dataframe using pandas's 'read_csv' function. The dataframe consists of the following data fields – Postal Code, Latitude and Longitude.

```
In [6]: df_coord = pd.read_csv('Geospatial_Coordinates.csv')
df_coord.head()
```

Out[6]:

	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

<u>Source 3: Information on each Neighbourhood in Toronto</u>

This set of information data is obtained through a request via API call from Foursquare. What was requested are features that are selected, which are namely – all the types of venues that are available from each neighborhood as well as the geographical locations of each of these venue.

Hence to do this, the neighbourhoods' name and geographical location (that can be found from the merged table from Source 1 and 2) is identified as the primary key and be used as a the data input required for the API request function defined as below:

```
In [14]: # Explore Neighborhoods in Toronto
import requests
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={}'.forr
            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 = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

The resultant table will return a dataframe of venue's details (*Venue, Venue Latitude, Venue Longitude, Venue Category*) along with its corresponding Neighborhood details (*Neighborhood, Neighborhood Longtitude, Neighborhood Latitude*)

3. Methodology

3.1 Data Manipulation

From Source 3's transformed data, we have selected the following features for John's main analysis requirement. An sample data set can be seen as below:

```
In [16]: print(toronto_venues.shape)
toronto_venues.head()

(1591, 7)

Out[16]:
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Regent Park, Harbourfront	43.65426	-79.360636	Tandem Coffee	43.653559	-79.361809	Coffee Shop
1	Regent Park, Harbourfront	43.65426	-79.360636	Roselle Desserts	43.653447	-79.362017	Bakery
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

To describe the dataframe, each row represent the details of a venue that is present in a given neighbourhood. In other words, we can simply do a count function on the dataframe (group by its respective neighborhoods), we can see that in every neighbourhood there are various types of venue, as shown below (note that what is illustrated is not the full list).

In [17]:
```
toronto_venues.groupby('Neighborhood').count()
```

Out[17]:

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Berczy Park	54	54	54	54	54	54
Brockton, Parkdale Village, Exhibition Place	22	22	22	22	22	22
Business reply mail Processing Centre, South Central Letter Processing Plant Toronto	16	16	16	16	16	16
CN Tower, King and Spadina, Railway Lands, Harbourfront West, Bathurst Quay, South Niagara, Island airport	16	16	16	16	16	16
Central Bay Street	61	61	61	61	61	61
Christie	16	16	16	16	16	16
Church and Wellesley	80	80	80	80	80	80
Commerce Court, Victoria Hotel	100	100	100	100	100	100
Davisville	34	34	34	34	34	34
Davisville North	8	8	8	8	8	8
Dufferin, Dovercourt Village	17	17	17	17	17	17
First Canadian Place, Underground city	100	100	100	100	100	100
Forest Hill North & West, Forest Hill Road Park	4	4	4	4	4	4
Garden District, Ryerson	100	100	100	100	100	100
Harbourfront East, Union Station, Toronto Islands	100	100	100	100	100	100
High Park, The Junction South	24	24	24	24	24	24
India Bazaar, The Beaches West	19	19	19	19	19	19

Next, I performed one hot encoded on dataframe: *toronto_venues* as part of data preparation process:

In [18]:
```
# one hot encoding
toronto_onehot = pd.get_dummies(toronto_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
toronto_onehot['Neighborhood'] = toronto_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [toronto_onehot.columns[-1]] + list(toronto_onehot.columns[:-1])
toronto_onehot = toronto_onehot[fixed_columns]

toronto_onehot.head()
```

Out[18]:

	Yoga Studio	Adult Boutique	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	American Restaurant	Antique Shop	...	Theme Restaurant	Tibetan Restaurant	Toy / Game Store	Trail	Train Station	Vegetarian / Vegan Restaurant
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0

5 rows × 233 columns

In [19]:
```
toronto_onehot.shape
```

Then, making use of this interim processed dataframe, we can create a new dataframe: *toronto_grouped,* that shows data of the frequency of occurrence of each venue category, per neighbourhood.

In [20]:
```python
#group rows by neighborhood and by taking the mean of the frequency of occurrence of each category
toronto_grouped = toronto_onehot.groupby('Neighborhood').mean().reset_index()
toronto_grouped
```

Out[20]:

	Neighborhood	Yoga Studio	Adult Boutique	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	American Restaurant	...	Theme Restaurant	Tibetan Restaurant	Toy / Game Store	Trail	T Sta
0	Berczy Park	0.000000	0.0000	0.0000	0.0000	0.0000	0.000	0.0000	0.0000	0.000000	...	0.0000	0.00000	0.000000	0.00000	
1	Brockton, Parkdale Village, Exhibition Place	0.000000	0.0000	0.0000	0.0000	0.0000	0.000	0.0000	0.0000	0.000000	...	0.0000	0.00000	0.000000	0.00000	
2	Business reply mail Processing Centre, South C...	0.000000	0.0000	0.0000	0.0000	0.0000	0.000	0.0000	0.0000	0.000000	...	0.0000	0.00000	0.000000	0.00000	
3	CN Tower, King and Spadina, Railway Lands, Har...	0.000000	0.0000	0.0625	0.0625	0.0625	0.125	0.1875	0.0625	0.000000	...	0.0000	0.00000	0.000000	0.00000	
	Central Bay...															

This is then followed by a series of data manipulation steps, which includes processes to rank the frequency of occurrence of each category within each neighborhood and then labelling from the venue with the highest frequency for each neighborhood as the *1st Most Common Venue* (I have only analysed up to the top 10 most common venues, since the interest here is looking at what is deemed as the most frequent establishments/venue which can be used as the final result as recommendation for John)

In [24]:
```python
# create the new dataframe and display the top 10 venues for each neighborhood
num_top_venues = 10

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = toronto_grouped['Neighborhood']

for ind in np.arange(toronto_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(toronto_grouped.iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted.head()
```

Out[24]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Berczy Park	Coffee Shop	Cocktail Bar	Beer Bar	Seafood Restaurant	Restaurant	Bakery	Farmers Market	Cheese Shop	Concert Hall	Bistro
1	Brockton, Parkdale Village, Exhibition Place	Café	Breakfast Spot	Coffee Shop	Convenience Store	Burrito Place	Italian Restaurant	Restaurant	Stadium	Intersection	Bar
2	Business reply mail Processing Centre, South C...	Light Rail Station	Park	Comic Shop	Brewery	Burrito Place	Farmers Market	Fast Food Restaurant	Restaurant	Recording Studio	Skate Park
3	CN Tower, King and Spadina, Railway Lands, Har...	Airport Service	Airport Lounge	Boat or Ferry	Boutique	Airport	Airport Food Court	Airport Gate	Airport Terminal	Sculpture Garden	Coffee Shop
4	Central Bay Street	Coffee Shop	Sandwich Place	Café	Italian Restaurant	Salad Place	Burger Joint	Bubble Tea Shop	Portuguese Restaurant	Poke Place	Ramen Restaurant

4. Results & Discussion

4.1 Main Analysis – The Venue with the most number of counts under the '1st Most Common Venue'.

From the final aggregated dataframe: *neighborhood_venues_sorted* , we can simply apply values count function to the '1st Most Common Venue' column to find out what is the venue category that is the most common. In which, the result showed that 'Coffee Shop' is has the highest number of counts in the column.

1st Most Common Venue	
Airport Service	1
Bar	1
Breakfast Spot	1
Business Service	1
Café	5
Clothing Store	1
Coffee Shop	16
Dessert Shop	1
Fast Food Restaurant	1
Greek Restaurant	1
Grocery Store	1
Health & Beauty Service	1
Light Rail Station	1
Park	3
Pharmacy	1
Sandwich Place	1
Summer Camp	1
Trail	1

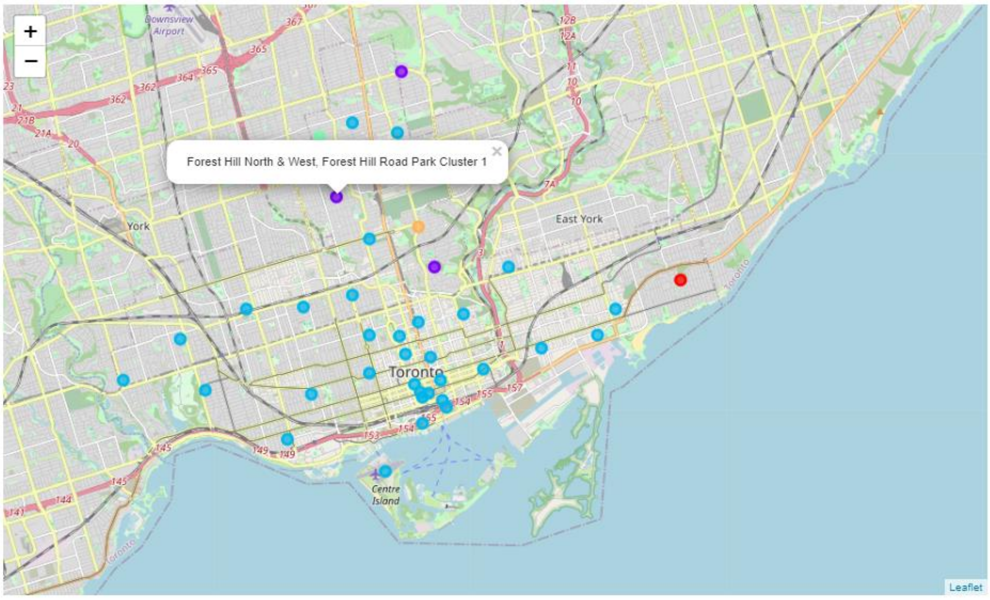
This result implies that across all the neighbourhoods, most have 'Coffee Shop' as their 1st Common Venue, which demonstrates its popularity in Toronto at the moment. As such, it can be recommended to John as the most 'prospective business' to embark on his entrepreneur career.

5. Conclusion

Based on the result computed in 4. Results and Discussion, John has identified coffee shop as the most probable business to start on.

Moving forward, as a future work for additional analysis, he may also want to find out which neighborhood has similar market trends so that he could better plan on how and where he should open these coffee shop outlets. For this retrospective, we could possibly perform K-Means Clustering on the neighborhoods to find out the similarities/dissimilarities amongst them. Since this is an additional piece of analysis, I will not be elaborating further in this report but I have included in my repository an example of how I have perform such clustering technique on the dataset, where I used K=5 as an example.

Out[38]:



Forest Hill North & West, Forest Hill Road Park Cluster 1
Downsview Airport
York
East York
Toronto
Centre Island
Leaflet