Most Prospective Business in Toronto

Capstone Project - The Battle of Neighborhoods (Week 2)

Jeffrey Chai Zhi Yang
9 February 2021

Table of Contents

Introduction (1/2)

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.

Business Problem

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.

Introduction (2/2)

+ 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.

Data

+ Data Sources & Pre-processing

Source 1

List of Postal Codes (only M) of Canada

Web-Scraping | Convert to dataframe

Source 2

Geographical coordinates of each postal code (only M) in Toronto

Read CSV | Convert to dataframe

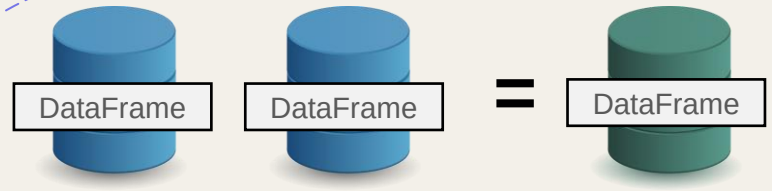
Source 3

Information on each Neighbourhood in Toronto

API Call | Convert to dataframe

Methodology

+Overview



Source 1 + Source 2 = “Toronto_data”

Inputs to function:
['Neighbourhood'],
['Latitude'],
['Longitude'],

API Call to Foursquare

Defined Function for API Call: *“getNearbyVenues”*

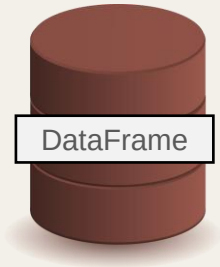
= DataFrame

“Toronto_venues”

+

A series of data manipulation steps…*

- On this final aggregated table, perform count function on the **‘1st Most Visited Venue’**# Column.
- The venue with the most number of counts of venue = **Most Prospective Business!**



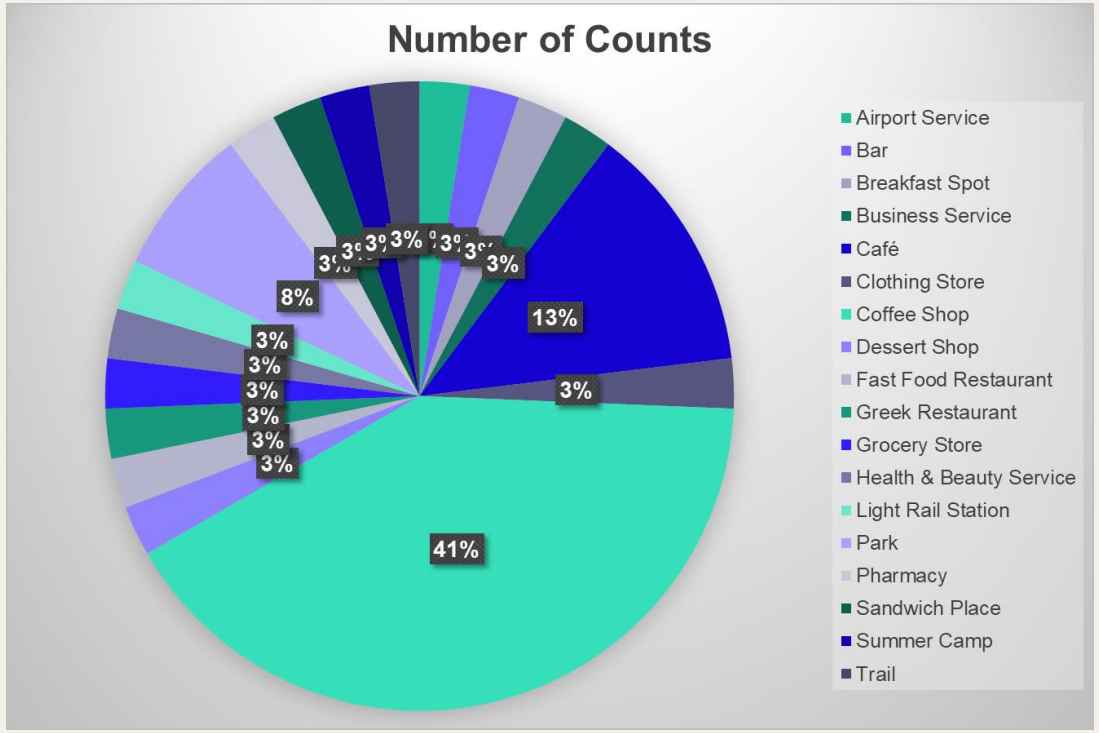
“neighborhoods_venues_sorted”

=

**For more details on the data manipulation steps, please refer to the Jupyter notebook in the repository*
 # One of the columns in ‘neighborhoods_venues_sorted’ dataframe.

Results & Discussion

+Most Number of Counts in ‘1st Most Visited Venue’



COFFEE SHOP (41%)

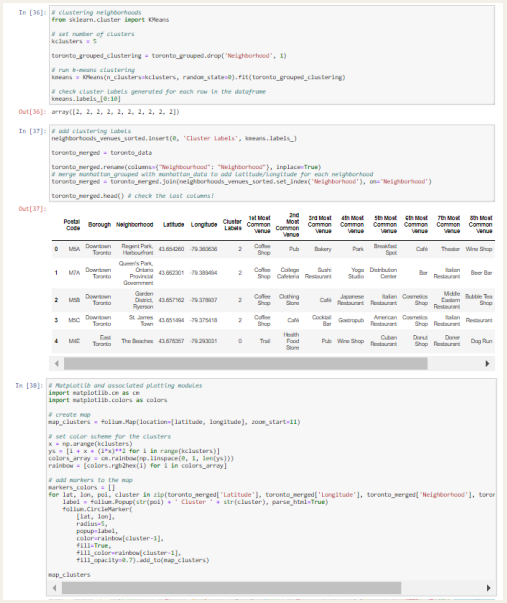
Conclusion

+ To recommend John to try ‘Coffee Shop’ for his first entrepreneur business!

+ Future Work for additional Analysis
  - Where should John better allocate his resources such that he can strategically set up his first set of coffee shop outlets?
  - How can he discover and understand more about the market trends of individual neighbourhoods?
  - Is he able to identify neighborhoods that generally loves coffee shops more? Or loves the similar common venues should John wish to venture into our sort of businesses in the future?

Future Work (Cont.)

+ Possible Data Science Method to find similar/dissimilar market trends across neighborhoods in Toronto – **K-Means Clustering^**



```
In [36]: # clustering neighborhoods
         from sklearn.cluster import KMeans

         # set number of clusters
         kclusters = 5

         toronto_grouped_clustering = toronto_grouped.drop('Neighborhood', 1)

         # run k-means clustering
         kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(toronto_grouped_clustering)

         # check cluster labels generated for each row in the dataframe
         kmeans.labels_[0:10]

Out[36]: array([2, 2, 2, 2, 2, 2, 2, 2, 2, 2])

In [37]: # add clustering labels
         neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

         toronto_merged = toronto_data

         toronto_merged.rename(columns={"Neighborhood": "Neighborhood"}, inplace=True)
         # merge manhattan_grouped with manhattan_data to add latitude/longitude for each neighborhood
         toronto_merged = toronto_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')

         toronto_merged.head() # check the last columns!

Out[37]:
```

	Postal Code	Borough	Neighborhood	Latitude	Longitude	Cluster Labels	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
0	M5A	Downtown Toronto	Regent Park, Harbourfront	43.654260	-79.360636	2	Coffee Shop	Pub	Bakery	Park	Breakfast Spot	Café	Theater	Wine Shop
1	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494	2	Coffee Shop	College Cafeteria	Sushi Restaurant	Yoga Studio	Distribution Center	Bar	Italian Restaurant	Beer Bar
2	M5B	Downtown Toronto	Garden District, Ryerson	43.657162	-79.378937	2	Coffee Shop	Clothing Store	Café	Japanese Restaurant	Italian Restaurant	Cosmetics Shop	Middle Eastern Restaurant	Bubble Tea Shop
3	M5C	Downtown Toronto	St. James Town	43.651494	-79.375418	2	Coffee Shop	Café	Cocktail Bar	Gastropub	American Restaurant	Cosmetics Shop	Italian Restaurant	Restaurant
4	M4E	East Toronto	The Beaches	43.676357	-79.293031	0	Trail	Health Food Store	Pub	Wine Shop	Cuban Restaurant	Donut Shop	Dorner Restaurant	Dog Run

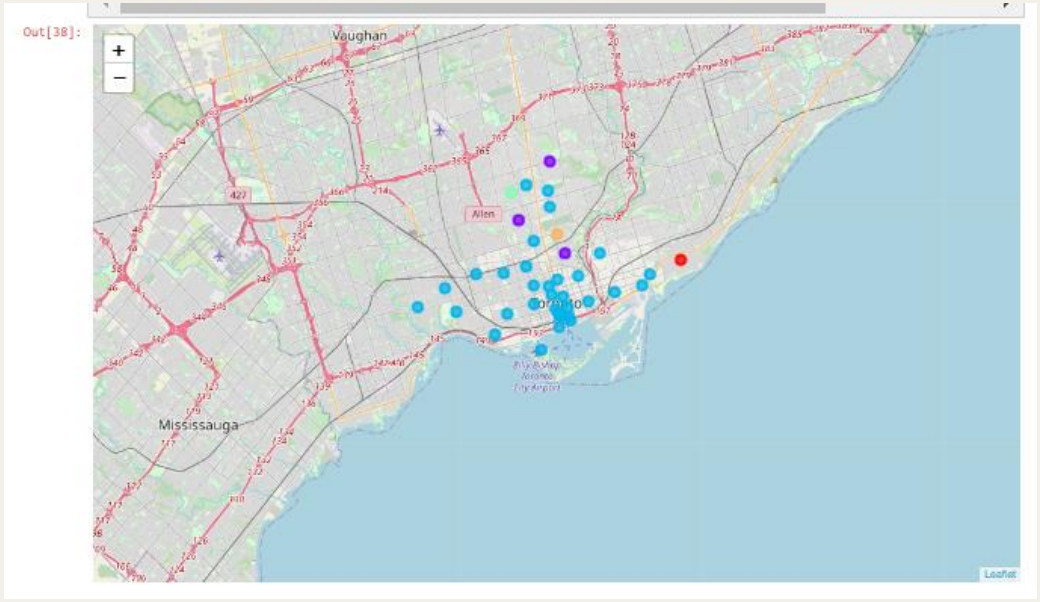
```
In [38]: # Matplotlib and associated plotting modules
         import matplotlib.cm as cm
         import matplotlib.colors as colors

         # create map
         map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

         # set color scheme for the clusters
         x = np.arange(kclusters)
         ys = [i + x + (i*x)**2 for i in range(kclusters)]
         colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
         rainbow = [colors.rgb2hex(i) for i in colors_array]

         # add markers to the map
         markers_colors = []
         for lat, lon, poi, cluster in zip(toronto_merged['Latitude'], toronto_merged['Longitude'], toronto_merged['Neighborhood'], toror
             label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
             folium.CircleMarker(
                 [lat, lon],
                 radius=5,
                 popup=label,
                 color=rainbow[cluster-1],
                 fill=True,
                 fill_color=rainbow[cluster-1],
                 fill_opacity=0.7).add_to(map_clusters)

         map_clusters
```



^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.