COURSERA CAPSTONE FINAL REPORT

April 4, 2020

Prepared by:
Mohammed Saidul Islam

Contents

1 BUSINESS PROBLEM DESCRIPTION

The business problem that is addressed in this notebook is that, **if a person wants to open a new coffee shop** in a city in Canada, then what are the things that he/she has to look into before opening the shop. Here, by analyzing and exploring all of the Neighborhoods in the **Boroughs(North York, East York and York)** in the city **Vaughan**, he can get useful insights about the **venues** present in the neighborhoods. If he/she can find a neighborhood where no coffee shop is present currently he/she could try to establish one in that neighborhood. Also, he/she has to explore the neighboring neighborhoods to get more better insights for his/her business.

In this case, the stakeholders are himself/herself and the people in the neighborhoods. As he/she will be the **owner** of the coffee shop, and he/she wants to make profit off of it, he/she needs to analyze all the neighborhoods near the city. So, he/she will be the **internal stakeholder**.

And the customer will be the consumers. The popularity and prosperity of his/her business will very much depend of the customers' mood, whether they like the coffee shop or not, whether they like the services given by the employees or not. So, the **customers** will be the **external stakeholder** of the business.

2 OVERVIEW OF THE DATASET

The **dataset** that I am working on is the **Neighborhood data of Canada** according to their **postal codes**. It has been downloaded from the wikipedia page: Canada Postal codes. To scrape the webpage, I have used the **"beautifulsoup4"** library. The dataset consists of **three columns**, namely, **PostalCode** ==>refers to the postal code of each of the Neighborhood, **Borough** ==>the Borough in which the Neighborhood is situated, and **Neighborhood** ==>the name of the Neighborhood. To explore each of the Neighborhoods, where all of the **coffee shops, parks, restaurants** and **other venues**, the **Foursquare API** has been used. To use the Foursquare API I needed the **latitude** and the **longitude** values of each of the Neighborhoods. The latitude and the longitude values are collected from this website.

3 METHODOLOGY

As the business problem revolves around opening a coffee shop in a neighborhood in city of Vaughan in Canada, at first step the relevant **boroughs** are selected. The boroughs are: **North York, East York and York**.

In the second step, **all the neighborhoods** that resides in the boroughs selected have been figured out. After that, using the **foursquare API**, the **venues** that are residing in those neighborhoods are found out.

In the next step, **filtering** of the neighborhoods have been done based on the criteria on the absence of coffee shops. This results in the neighborhoods in those boroughs that does not have any coffee shops in them.

Finally, a **clustering technique (k-means clustering)** was used to find the clusters of similar neighborhoods. The clustering gives the necessary insight that is needed to find a place where if the coffee shop is established would result in **higher profit and customer satisfaction** for the owner.

4 ANALYSIS ON THE DATA

At first the selected neighborhoods are one hot encoded based on the data collected from the foursquare API.

Part 06: Analysis on the neighborhoods without any Coffee shops

In [44]:
```
# one hot encoding
neigh_no_coffeeShops_onehot = pd.get_dummies(neigh_no_coffeeShops[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
neigh_no_coffeeShops_onehot['Neighborhood'] = neigh_no_coffeeShops['Neighborhood']

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

neigh_no_coffeeShops_onehot.head()
```

Out[44]:

	Neighborhood	Airport	Asian Restaurant	Athletics & Sports	Bakery	Bank	Baseball Field	Basketball Court	Beer Store	Brewery	...	Pet Store	Pharmacy	Pizza Place	Pool	Pub	Shopping Mall	Skating Rink	Trail	Video Store	Women's Store
0	Parkwoods	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	Parkwoods	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	Parkview Hill / Woodbine Gardens	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	Parkview Hill / Woodbine Gardens	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	Parkview Hill / Woodbine Gardens	0	0	0	0	0	0	0	0	0	...	0	1	0	0	0	0	0	0	0	0

5 rows × 46 columns

Figure 1: One hot encoded data

Then the neighborhoods are grouped by the mean of the one-hot values.

Grouping rows by neighborhood and by taking the mean of the frequency of occurrence of each category

```
In [46]: neigh_no_coffeeShops_grouped = neigh_no_coffeeShops_onehot.groupby('Neighborhood').mean().reset_index()
         neigh_no_coffeeShops_grouped
```

Out[46]:

	Neighborhood	Airport	Asian Restaurant	Athletics & Sports	Bakery	Bank	Baseball Field	Basketball Court	Beer Store	Brewery	...	Pet Store	Pharmacy	Pizza Place	Pool	Pub	Shopping Mall	Skating Rink	T
0	Bayview Village	0.000000	0.000000	0.000000	0.00	0.250000	0.000000	0.00	0.000000	0.00	...	0.000000	0.000000	0.000000	0.0	0.00	0.000000	0.000000	0.000
1	Caledonia-Fairbanks	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.00	...	0.000000	0.000000	0.000000	0.0	0.00	0.000000	0.000000	0.000
2	Downsview	0.071429	0.000000	0.071429	0.00	0.071429	0.071429	0.00	0.000000	0.00	...	0.000000	0.000000	0.000000	0.0	0.00	0.071429	0.000000	0.000
3	Glencairn	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.00	...	0.000000	0.000000	0.250000	0.0	0.25	0.000000	0.000000	0.000
4	Hillcrest Village	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.00	...	0.000000	0.000000	0.000000	0.2	0.00	0.000000	0.000000	0.000
5	Humber Summit	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.00	...	0.000000	0.000000	0.000000	0.0	0.00	0.000000	0.000000	0.000
6	Humberlea / Emery	0.000000	0.000000	0.000000	0.00	0.000000	1.000000	0.00	0.000000	0.00	...	0.000000	0.000000	0.000000	0.0	0.00	0.000000	0.000000	0.000
7	Humewood-Cedarvale	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.00	...	0.000000	0.000000	0.000000	0.0	0.00	0.000000	0.000000	0.333
8	North Park / Maple Leaf Park / Upwood Park	0.000000	0.000000	0.000000	0.25	0.000000	0.000000	0.25	0.000000	0.00	...	0.000000	0.000000	0.000000	0.0	0.00	0.000000	0.000000	0.000
9	Parkview Hill / Woodbine Gardens	0.000000	0.000000	0.090909	0.00	0.090909	0.000000	0.00	0.000000	0.00	...	0.090909	0.090909	0.181818	0.0	0.00	0.000000	0.000000	0.000

Figure 2: Group by data

In the next step, top five venues of each neighborhoods are generated.

Finding each neighborhood along with the top 5 most common venues

```
In [48]: num_top_venues = 5

         for hood in neigh_no_coffeeShops_grouped['Neighborhood']:
             print("----"+hood+"----")
             temp = neigh_no_coffeeShops_grouped[neigh_no_coffeeShops_grouped['Neighborhood'] == hood].T.reset_index()
             temp.columns = ['venue','freq']
             temp = temp.iloc[1:]
             temp['freq'] = temp['freq'].astype(float)
             temp = temp.round({'freq': 2})
             print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_venues))
             print('\n')
```

```
----Woodbine Heights----
              venue  freq
0      Skating Rink  0.18
1      Dance Studio  0.09
2          Pharmacy  0.09
3  Athletics & Sports  0.09
4       Video Store  0.09


----York Mills West----
              venue  freq
0              Park  0.50
1              Bank  0.25
2  Convenience Store  0.25
3           Airport  0.00
4     Grocery Store  0.00
```

Figure 3: Top 5 venues in neighborhoods

In the following step, the neighborhoods are merged and the venues are sorted in descending order based on their frequency values.

Sorting the venues in descending order

```
In [49]: def return_most_common_venues(row, num_top_venues):
             row_categories = row.iloc[1:]
             row_categories_sorted = row_categories.sort_values(ascending=False)

             return row_categories_sorted.index.values[0:num_top_venues]
```

```
In [50]: num_top_venues = 10

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

         # creating 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))

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

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

         neighborhoods_venues_sorted.head()
```

Out[50]:

	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	Bayview Village	Café	Chinese Restaurant	Bank	Japanese Restaurant	Empanada Restaurant	Dog Run	Discount Store	Dance Studio	Curling Ice	Cosmetics Shop

Figure 4: Sorted venues

Then, a clustering technique is used to cluster the neighborhoods and a map is generated based on the clusters.

Part 07: Clustering the Neighborhoods

Running K-means clustering algorithm to cluster the neighborhoods

```
In [51]: # setting number of clusters
         kclusters = 5

         neigh_no_coffeeShops_grouped_clustering = neigh_no_coffeeShops_grouped.drop('Neighborhood', 1)

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

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

Out[51]: array([0, 3, 0, 0, 0, 2, 1, 4, 0, 0], dtype=int32)
```

```
In [52]: # adding clustering labels
         neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

         neigh_no_coffeeShops_merged = neigh_no_coffeeShops

         # merging neigh_no_coffeeShops_grouped with neigh_no_coffeeShops to add latitude/longitude for each neighborhood
         neigh_no_coffeeShops_merged = neigh_no_coffeeShops_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')

         neigh_no_coffeeShops_merged.head()
```

Figure 5: Clustering the neighborhoods

Out[52]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category	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 Mos Commo Venu
0	Parkwoods	43.753259	-79.329656	Brookbanks Park	43.751976	-79.332140	Park	3	Food & Drink Shop	Park	Café	Empanada Restaurant	Dog Run	Discount Store	Dance Studio	Curlin Ic
1	Parkwoods	43.753259	-79.329656	Variety Store	43.751974	-79.333114	Food & Drink Shop	3	Food & Drink Shop	Park	Café	Empanada Restaurant	Dog Run	Discount Store	Dance Studio	Curlin Ic
2	Parkview Hill / Woodbine Gardens	43.706397	-79.309937	Jawny Bakers	43.705783	-79.312913	Gastropub	0	Pizza Place	Bus Line	Fast Food Restaurant	Athletics & Sports	Bank	Pharmacy	Pet Store	Gastropul
3	Parkview Hill / Woodbine Gardens	43.706397	-79.309937	East York Gymnastics	43.710654	-79.309279	Gym / Fitness Center	0	Pizza Place	Bus Line	Fast Food Restaurant	Athletics & Sports	Bank	Pharmacy	Pet Store	Gastropul
4	Parkview Hill / Woodbine Gardens	43.706397	-79.309937	Shoppers Drug Mart	43.705933	-79.312825	Pharmacy	0	Pizza Place	Bus Line	Fast Food Restaurant	Athletics & Sports	Bank	Pharmacy	Pet Store	Gastropul

Figure 6: Result of the clustering

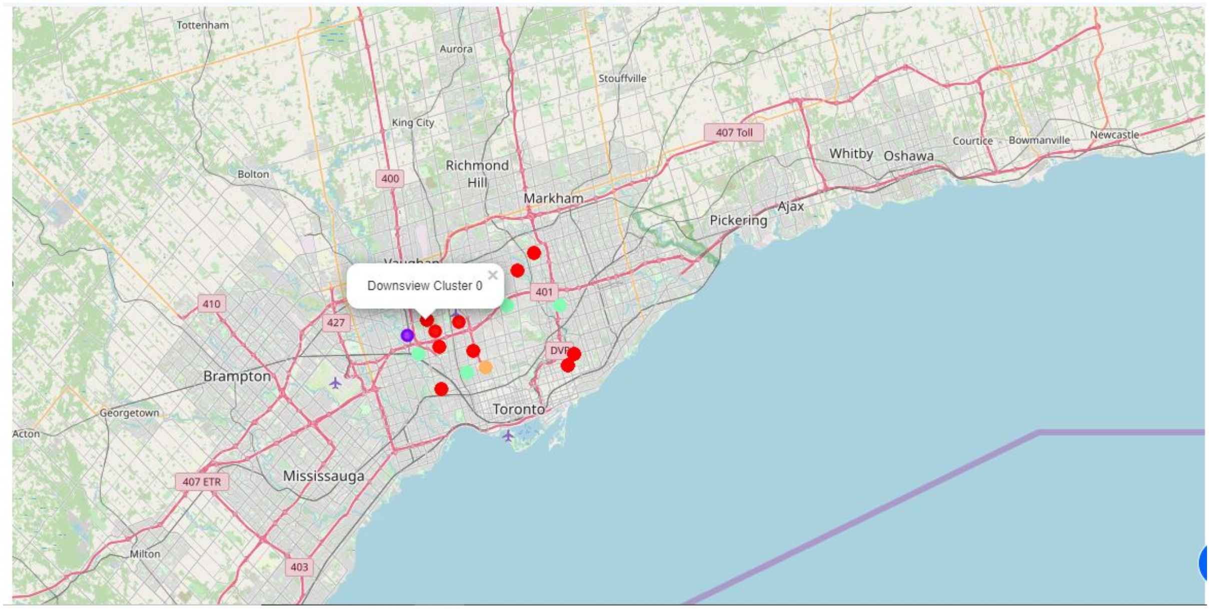

Figure 7: Map showing different neighborhood clusters

Finally, the clusters are examined and based on the result, clusters of neighborhoods are chosen based on their business potential.

Part 08: Examining the clusters

Cluster 1

```
In [54]: neigh_no_coffeeShops_merged.loc[neigh_no_coffeeShops_merged['Cluster Labels'] == 0, neigh_no_coffeeShops_merged.columns[[1] + list(range(5, neigh_no_cof
```

Out[54]:

	Neighborhood Latitude	Venue Longitude	Venue Category	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	9th Most Common Venue	10th Most Common Venue
2	43.706397	-79.312913	Gastropub	0	Pizza Place	Bus Line	Fast Food Restaurant	Athletics & Sports	Bank	Pharmacy	Pet Store	Gastropub	Gym / Fitness Center	Intersection
3	43.706397	-79.309279	Gym / Fitness Center	0	Pizza Place	Bus Line	Fast Food Restaurant	Athletics & Sports	Bank	Pharmacy	Pet Store	Gastropub	Gym / Fitness Center	Intersection
4	43.706397	-79.312825	Pharmacy	0	Pizza Place	Bus Line	Fast Food Restaurant	Athletics & Sports	Bank	Pharmacy	Pet Store	Gastropub	Gym / Fitness Center	Intersection
5	43.706397	-79.312270	Bank	0	Pizza Place	Bus Line	Fast Food Restaurant	Athletics & Sports	Bank	Pharmacy	Pet Store	Gastropub	Gym / Fitness Center	Intersection
6	43.706397	-79.313130	Pizza Place	0	Pizza Place	Bus Line	Fast Food Restaurant	Athletics & Sports	Bank	Pharmacy	Pet Store	Gastropub	Gym / Fitness Center	Intersection

Figure 8: Cluster 1 data

Cluster 2

```
In [55]: neigh_no_coffeeShops_merged.loc[neigh_no_coffeeShops_merged['Cluster Labels'] == 1, neigh_no_coffeeShops_merged.columns[[1] + list(range(5, neigh_no_coffe
```

Out[55]:

	Neighborhood Latitude	Venue Longitude	Venue Category	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	9th Most Common Venue	10th Most Common Venue
63	43.724766	-79.532854	Baseball Field	1	Baseball Field	Women's Store	Chinese Restaurant	Empanada Restaurant	Dog Run	Discount Store	Dance Studio	Curling Ice	Cosmetics Shop	Convenience Store

Figure 9: Cluster 2 data

Cluster 3

```
In [56]: neigh_no_coffeeShops_merged.loc[neigh_no_coffeeShops_merged['Cluster Labels'] == 2, neigh_no_coffeeShops_merged.columns[[1] + list(range(5, neigh_no_coffe
```

Out[56]:

	Neighborhood Latitude	Venue Longitude	Venue Category	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	9th Most Common Venue	10th Most Common Venue
62	43.756303	-79.570637	Empanada Restaurant	2	Empanada Restaurant	Women's Store	Field	Dog Run	Discount Store	Dance Studio	Curling Ice	Cosmetics Shop	Convenience Store	Construction & Landscaping

Figure 10: Cluster 3 data

Cluster 4

```
In [57]: d.loc[neigh_no_coffeeShops_merged['Cluster Labels'] == 3, neigh_no_coffeeShops_merged.columns[[1] + list(range(5, neigh_no_coffeeShops_merged.shape[1]))]]
```

Out[57]:

	Neighborhood Latitude	Venue Longitude	Venue Category	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	9th Most Common Venue	10th Most Common Venue
0	43.753259	-79.332140	Park	3	Food & Drink Shop	Park	Café	Empanada Restaurant	Dog Run	Discount Store	Dance Studio	Curling Ice	Cosmetics Shop	Convenience Store
1	43.753259	-79.333114	Food & Drink Shop	3	Food & Drink Shop	Park	Café	Empanada Restaurant	Dog Run	Discount Store	Dance Studio	Curling Ice	Cosmetics Shop	Convenience Store
31	43.689026	-79.456300	Park	3	Park	Women's Store	Market	Dog Run	Discount Store	Dance Studio	Curling Ice	Cosmetics Shop	Convenience Store	Construction & Landscaping
32	43.689026	-79.456333	Women's Store	3	Park	Women's Store	Market	Dog Run	Discount Store	Dance Studio	Curling Ice	Cosmetics Shop	Convenience Store	Construction & Landscaping
33	43.689026	-79.456317	Market	3	Park	Women's Store	Market	Dog Run	Discount Store	Dance Studio	Curling Ice	Cosmetics Shop	Convenience Store	Construction & Landscaping

Figure 11: Cluster 4 data

Cluster 5

```
In [58]: neigh_no_coffeeShops_merged.loc[neigh_no_coffeeShops_merged['Cluster Labels'] == 4, neigh_no_coffeeShops_merged.columns[[1] + list(range(5, neigh_no_coffe
```

Out[58]:

	Neighborhood Latitude	Venue Longitude	Venue Category	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	9th Most Common Venue	10th Most Common Venue
28	43.693781	-79.428705	Field	4	Field	Trail	Hockey Arena	Chinese Restaurant	Empanada Restaurant	Dog Run	Discount Store	Dance Studio	Curling Ice	Cosmetics Shop
29	43.693781	-79.426106	Trail	4	Field	Trail	Hockey Arena	Chinese Restaurant	Empanada Restaurant	Dog Run	Discount Store	Dance Studio	Curling Ice	Cosmetics Shop
30	43.693781	-79.431761	Hockey Arena	4	Field	Trail	Hockey Arena	Chinese Restaurant	Empanada Restaurant	Dog Run	Discount Store	Dance Studio	Curling Ice	Cosmetics Shop

Figure 12: Cluster 5 data

From our cluster analysis, we can see that the neighborhoods that falls in **cluster 0** and **cluster 3** has more venues in them than the other clusters. So, those neighborhoods might have more **potential customers** for any business.

```
In [62]: #Finding out the neighborhoods of interest
         neighborhoods_of_interest_1 = neigh_no_coffeeShops_merged[neigh_no_coffeeShops_merged['Cluster Labels'] == 0].Neighborhood
         neighborhoods_of_interest_2 = neigh_no_coffeeShops_merged[neigh_no_coffeeShops_merged['Cluster Labels'] == 3].Neighborhood

In [65]: #Neighborhoods of interest: 01
         print(neighborhoods_of_interest_1.unique())

         ['Parkview Hill / Woodbine Gardens' 'Glencairn' 'Woodbine Heights'
          'Hillcrest Village' 'Bayview Village' 'Downsview'
          'North Park / Maple Leaf Park / Upwood Park'
          'Runnymede / The Junction North']

In [66]: #Neighborhoods of interest: 02
         print(neighborhoods_of_interest_2.unique())

         ['Parkwoods' 'Caledonia-Fairbanks' 'Weston' 'York Mills West']
```

Figure 13: Potential clusters of neighborhoods

5 Results and Discussion

So the cluster analysis results in 5 clusters of neighborhoods present in the boroughs of: North York, East York and York. To select the neighborhoods that would be perfect for opening a coffee shop two neighborhoods clusters have been selected, namely **cluster 0** and **cluster 3**.
In cluster 0, the neighborhoods present are: 'Parkview Hill / Woodbine Gardens', 'Glencairn', 'Woodbine Heights', 'Hillcrest Village', 'Bayview Village', 'Downsview', 'North Park / Maple Leaf Park / Upwood Park', 'Runnymede / The Junction North'.
In cluster 3, the neighborhoods present are: 'Parkwoods', 'Caledonia-Fairbanks', 'Weston', 'York Mills West'.
Although they fall in the same cluster, the distance between neighborhoods in cluster 3 is much greater than the neighborhoods in cluster 0.
So neighborhoods in cluster 0 would be a good choice for a potential neighborhood to open a coffee shop based on business perspective. Remember, the data that have been worked on, consists only of the neighborhoods that does not have any coffee shops in them. From the map analysis of the clusters it is found that the **Downsview** neighborhood might be the best choice in cluster 0.

6 Conclusion

Although the dataset consists of neighborhood data of every city in Canada and the foursquare API has been used to find out all the venues residing in those neighborhoods, but lack of population data, population density data in the neighborhoods certainly limit the capability to get a proper analysis of the business potential of each neighborhood. But, based on the current data, it can be said that, **Downsview** is a good choice to open a coffee shop in the city of Vaughan.