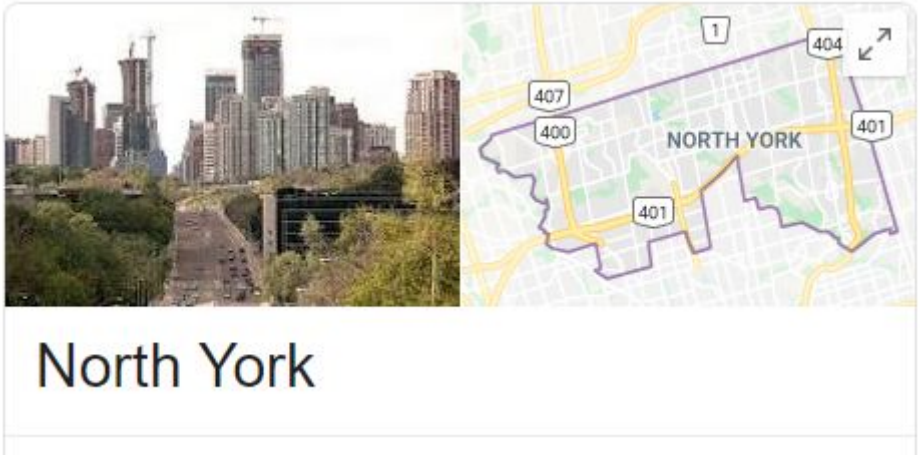
Introduction

Problem Description

There is a groceries contractor in one of the boroughs of Toronto (North York). This contractor provides places such as: Different types of Restaurants, Bakery, Breakfast Spot, Brewery and Café with fresh and high-quality groceries. The contractor wants to build a warehouse for the groceries it buys from villagers and farmers inside the borough, so that they will support more customers and also bring better "Quality of Service" to the old customers.

For example, if the warehouse is close to those old and famous restaurants, then the vegetables and other groceries would be delivered to the restaurant in the right time and there would be no delay so the restaurant cooks can start their job from the morning and the Quality of Service will be high and this contractor will gain more reputation and income.

The contractor should build this warehouse where it is closest to its customers in order to minimize the cost of transportation in addition to the example above. which neighborhood (in that borough) would be a better choice for the contractor to build the warehouse in that neighborhood. Finding the right neighborhood is our mission and our recommender system will provide this contractor with a sorted list of neighborhoods in which the first element of the list will be the best suggested neighborhood.



Data We Need

We will need geo-locational information about that specific borough and the neighborhoods in that borough. We specifically and technically mean the latitude and longitude numbers of that borough. We assume that it is "North York" in Toronto. This is easily provided for us by the contractor, because the contractor has already made up his mind about the borough. The Postal Codes that fall into that borough would also be sufficient for us. I fact we will first find neighborhoods inside North York by their corresponding Postal Codes.

We will need data about different venues in different neighborhoods of that specific borough. In order to gain that information we will use "Foursquare" locational information. By locational information for each venue we mean basic and advanced information about that venue. For example there is a venue in one of the neighborhoods. As basic information, we can obtain its precise latitude and longitude and also its distance from the center of the neighborhood. But we are looking for advanced information such as the category of that venue and whether this venue is a popular one in its category or maybe the average price of the services of this venue.

Methodology

Identifying Neighborhoods inside "North York"

We will use Postal Codes of different regions inside North York to find the list of neighborhoods. We will essentially obtain our information from https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M and then process the table inside this site. Images from dataframes and also from maps will be provided in the presentation. Here we only present our strategy and how we got the mission accomplished.

Connecting to Foursquare and Retrieving Locational Data for Each Venue in Every Neighborhood

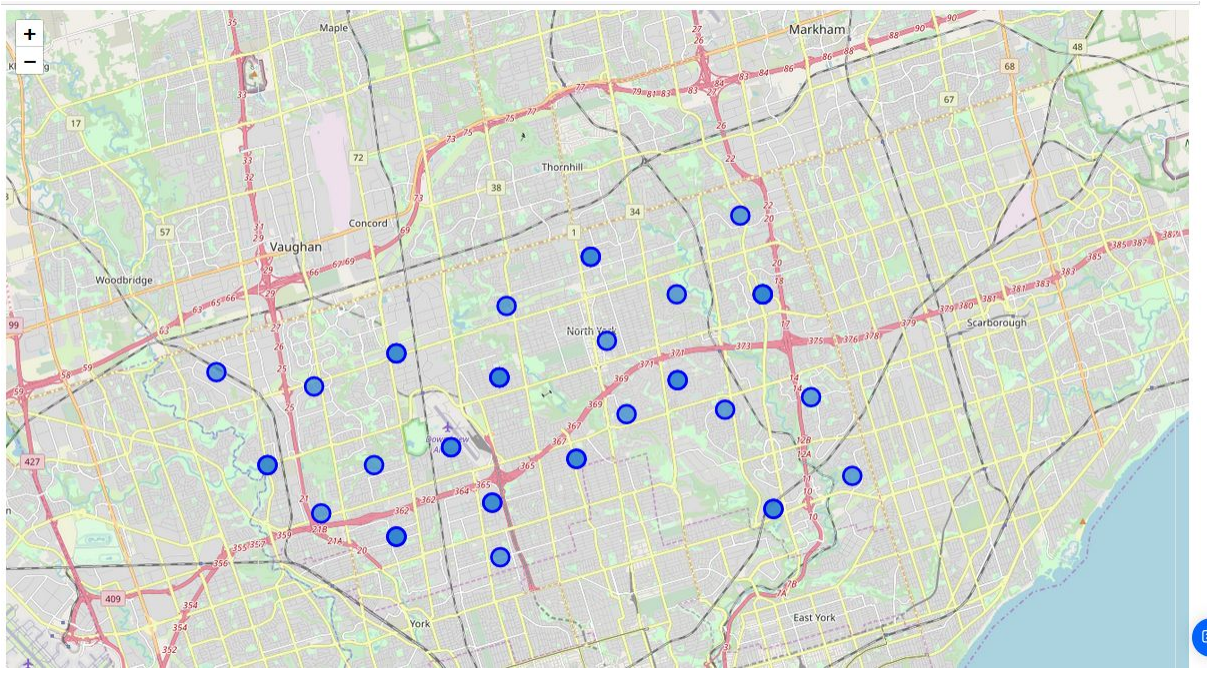
Focusing on the "North York" Borough in Toronto

```
In [5]: # selecting only neighborhoods regarding to "North York" borough.
north_york_data = neighborhood_df[neighborhood_df['Borough'] == 'North York']
north_york_data.head()
```

Out[5]:

	PostalCode	Borough	Neighborhood	Latitude	Longitude
0	M3A	North York	Parkwoods	43.752420	-79.329242
1	M4A	North York	Victoria Village	43.730600	-79.313265
3	M6A	North York	Lawrence Heights	43.723270	-79.451286
4	M6A	North York	Lawrence Manor	43.723270	-79.451286
9	M3B	North York	Don Mills North	43.749055	-79.362227

After finding the list of neighborhoods, we then connect to the Foursquare API to gather information about venues inside each and every neighborhood. For each neighborhood, we have chosen the radius to be 1000 meter. It means that we have asked Foursquare to find venues that are at most 1000 meter away from the center of the neighborhood. (I think distance is measured by latitude and longitude of venues and neighborhoods, and it is not the walking distance for venues.)



Data Wrangling

When the data is completely gathered, we will perform processing on that raw data to find our desirable features for each venue. Our main feature is the category of that venue. After this stage, the column "Venue's Category" will be One-hot encoded and different venues will have different feature-columns. After One-hot encoding we will integrate all restaurant columns to one column "Total Restaurants" and all food joint columns to "Total Joints" column. We assumed that different restaurants use the Same raw groceries. This assumption is made for simplicity and due to not having a very detailed dataset about different venues.

Displaying Venues for Each Neighborhood in North York

```
north_york_venues.head()
```

	Postal Code	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Summary	Venue Category	Distance
0	M3A	Parkwoods	43.75242	-79.329242	Brookbanks Park	This spot is popular	Park	238
1	M3A	Parkwoods	43.75242	-79.329242	Variety Store	This spot is popular	Food & Drink Shop	315
2	M4A	Victoria Village	43.73060	-79.313265	Wigmore Park	This spot is popular	Park	206
3	M4A	Victoria Village	43.73060	-79.313265	Memories of Africa	This spot is popular	Grocery Store	450
4	M4A	Victoria Village	43.73060	-79.313265	Guardian Drug	This spot is popular	Pharmacy	469

Now, the dataset is fully ready to be used for machine learning (and statistical analysis) purposes.

Applying one of Machine Learning Techniques (K-Means Clustering)

Here we cluster neighborhoods via K-means clustering method. We think that 5 clusters is enough and can cover the complexity of our problem. After clustering we will update our dataset and create a column representing the group for each neighborhood.

Run k-means to Cluster Neighborhoods into 5 Clusters

```
# import k-means from clustering stage
from sklearn.cluster import KMeans

# run k-means clustering
kmeans = KMeans(n_clusters = 5, random_state = 0).fit(north_york_onehot)
```

Decision Making

Now, we focus on the centers of clusters and compare them for their "Total Restaurants" and their "Total Joints". The group which its center has the highest "Total Sum" will be our best recommendation to the contractor. {Note: Total Sum = Total Restaurants + Total Joints + Other Venues.} This algorithm although is pretty straightforward yet is strongly powerful.

Showing Centers of Each Cluster ¶

```
means_df = pd.DataFrame(kmeans.cluster_centers_)
means_df.columns = north_york_onehot.columns
means_df.index = ['G1','G2','G3','G4','G5']
means_df['Total Sum'] = means_df.sum(axis = 1)
means_df.sort_values(axis = 0, by = ['Total Sum'], ascending=False)
```

]:

	Burrito Place	Café	Coffee Shop	Food & Drink Shop	Food Court	Grocery Store	Juice Bar	Kitchen Supply Store	Market	Pizza Place	Sandwich Place	Smoothie Shop	Supermarket	Tea Room	Total Restaurants	Total Joints	Total Sum
	1.000000e+00	5.551115e-17	3.000000	0.000000	2.000000	0.000000	1.000000e+00	0.000000e+00	0.000000e+00	0.000000	0.000000e+00	1.000000e+00	0.000000	2.000000e+00	6.000000	2.000000	18.000000
	0.000000e+00	2.000000e+00	1.800000	0.000000	0.000000	0.600000	6.000000e-01	0.000000e+00	2.000000e-01	0.600000	8.000000e-01	0.000000e+00	0.400000	0.000000e+00	8.600000	0.600000	16.800000
	0.000000e+00	1.000000e+00	2.000000	0.000000	1.000000	0.000000	0.000000e+00	1.000000e+00	0.000000e+00	0.000000	0.000000e+00	1.000000e+00	0.000000	1.000000e+00	4.000000	2.000000	14.000000
	0.000000e+00	0.000000e+00	0.750000	0.000000	0.000000	0.750000	0.000000e+00	0.000000e+00	0.000000e+00	1.750000	2.500000e-01	0.000000e+00	0.000000	0.000000e+00	4.750000	0.250000	9.000000
	2.775558e-17	1.110223e-16	0.526316	0.052632	0.105263	0.210526	5.551115e-17	3.469447e-17	1.734723e-17	0.052632	2.775558e-17	2.775558e-17	0.105263	1.387779e-16	0.105263	0.052632	1.578947

Results:

Based on this analysis, the best recommended neighborhoods will be:

{'Neighborhood': 'Fairview\n',
 'Neighborhood Latitude': 43.780970000000025,
 'Neighborhood Longitude': -79.34781328099996}

{'Neighborhood': 'Henry Farm',
 'Neighborhood Latitude': 43.780970000000025,
 'Neighborhood Longitude': -79.34781328099996}

{'Neighborhood': 'Oriole\n',
 'Neighborhood Latitude': 43.780970000000025,
 'Neighborhood Longitude': -79.34781328099996}

Conclusion:

Thus the grocery dealer who opens the dealership in these three areas will profit a lot when compared to the rest of the area as the opportunity of selling to different vendors are more in these locations.