IBM Data Science Capstone Project – Determining the best Grocery store in Downtown Toronto

Prepared by: Muhammad Sajjad Ashraf

Introduction:

Many people who live in urban and suburban areas will find a substantial grocery store count near their homes and commutes. The question then becomes: How does one decide which grocery store to shop at among all the options? There are many different criteria to use – convenience, variety, price and so forth – so how do you know which store will hit that sweet spot? The problem with the defined criteria is that one store may be better in some of the criteria set while others may have an advantage in any other. So, the mystery to choose the right Grocery store becomes complicated.

This is where data science & machine learning kicks in & provide a quick suggestion on the best available option for the grocery store.

Problem:

To determine which neighborhood has the best Grocery store in Downtown Toronto. Based on the output from foursquare API, user can easily find out which Grocery Store is best to visit based on feedback.

Data Requirements:

For this problem, I will be utilizing the Foursquare API to pull the following location data on Grocery Stores in Downtown Toronto.

- Venue Name
- Venue ID
- Venue Location
- Venue Category
- Total Likes

Data acquisition and cleaning:

Data Link: https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M

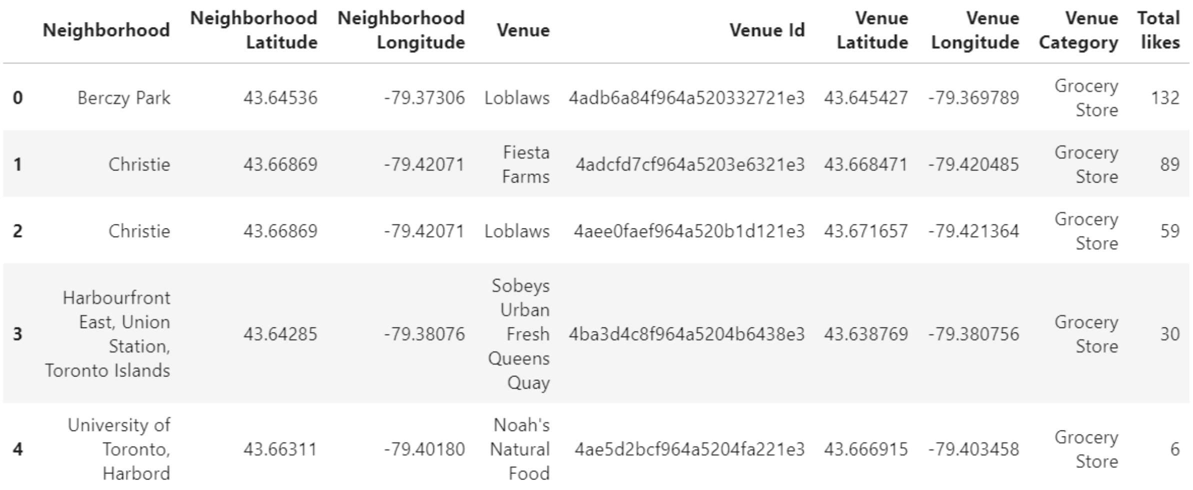
Will use Downtown Toronto dataset which we scrapped from wikipedia. Dataset consisting of latitude and longitude, zip codes.

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. Foursquare is a location data provider with information about all manner of venues and events within an area of interest. Such information includes venue names, locations, menus and even photos. As such, the foursquare location platform will be used as the sole data source since all the stated required information can be obtained through the API.

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 100 meter.

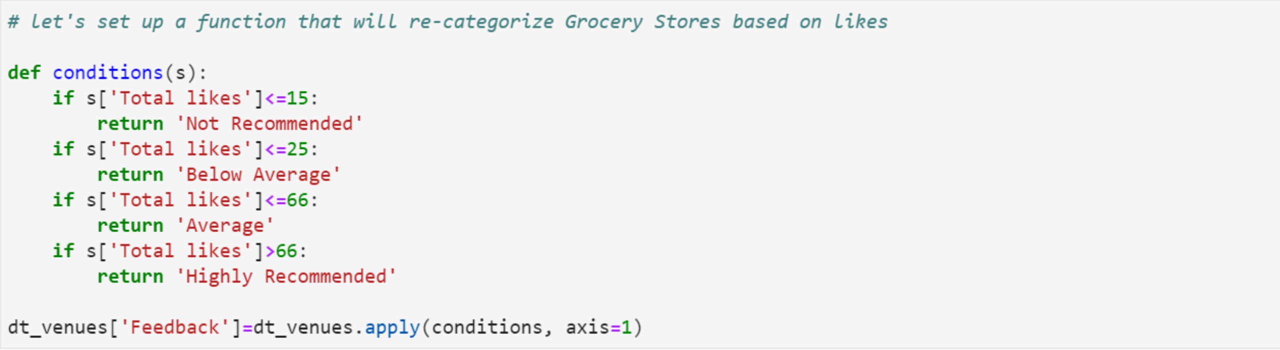
The data retrieved from Foursquare contained information of venues within a specified distance of the longitude and latitude of the postcodes. This data was used to create a pandas data frame. The information obtained per venue as follows:

1. Neighborhood
2. Neighborhood Latitude
3. Neighborhood Longitude
4. Venue
5. Venue ID
6. Venue Latitude
7. Venue Longitude
8. Venue Category
9. Total Likes

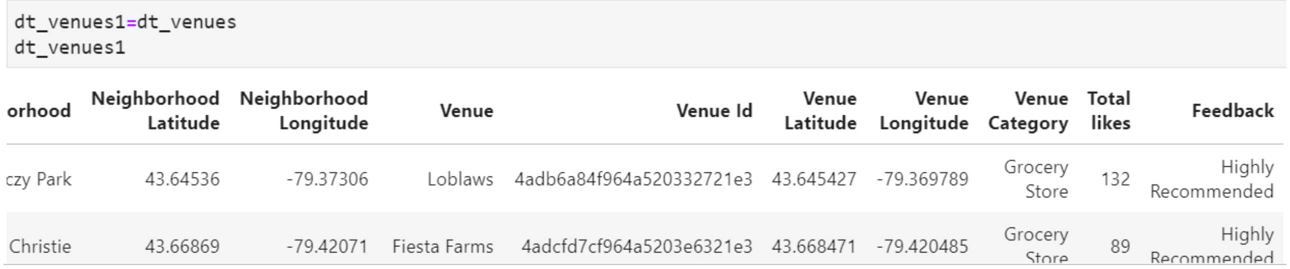


	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Id	Venue Latitude	Venue Longitude	Venue Category	Total likes
0	Berczy Park	43.64536	-79.37306	Loblaws	4adb6a84f964a520332721e3	43.645427	-79.369789	Grocery Store	132
1	Christie	43.66869	-79.42071	Fiesta Farms	4adcfd7cf964a5203e6321e3	43.668471	-79.420485	Grocery Store	89
2	Christie	43.66869	-79.42071	Loblaws	4aee0faef964a520b1d121e3	43.671657	-79.421364	Grocery Store	59
3	Harbourfront East, Union Station, Toronto Islands	43.64285	-79.38076	Sobeys Urban Fresh Queens Quay	4ba3d4c8f964a5204b6438e3	43.638769	-79.380756	Grocery Store	30
4	University of Toronto, Harbord	43.66311	-79.40180	Noah's Natural Food	4ae5d2bcf964a5204fa221e3	43.666915	-79.403458	Grocery Store	6

Now create a function that will re-categorize Grocery Stores based on Total likes.



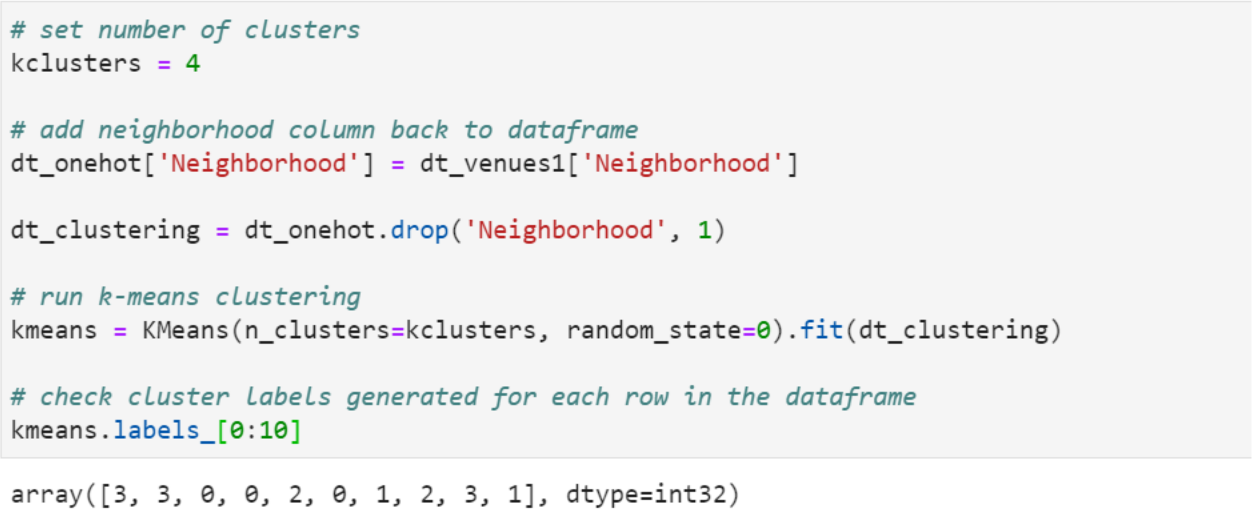
```
# let's set up a function that will re-categorize Grocery Stores based on likes

def conditions(s):
    if s['Total likes']<=15:
        return 'Not Recommended'
    if s['Total likes']<=25:
        return 'Below Average'
    if s['Total likes']<=66:
        return 'Average'
    if s['Total likes']>66:
        return 'Highly Recommended'

dt_venues['Feedback']=dt_venues.apply(conditions, axis=1)
```



```
dt_venues1=dt_venues
dt_venues1
```

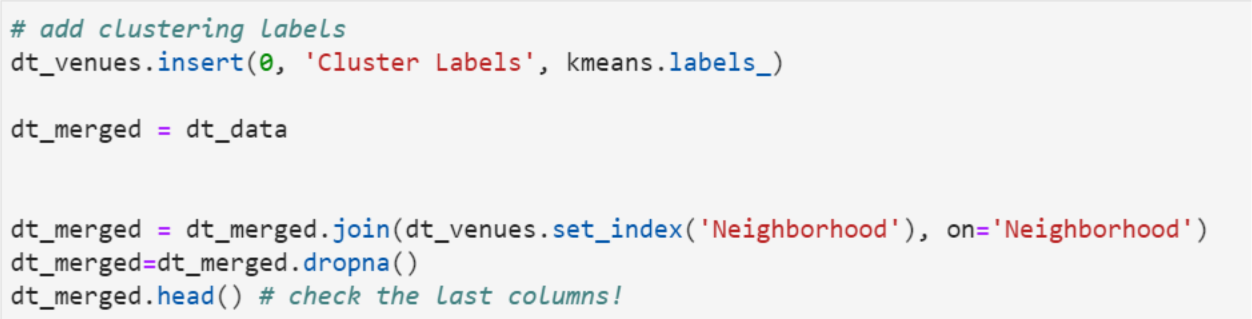
orhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Id	Venue Latitude	Venue Longitude	Venue Category	Total likes	Feedback
czy Park	43.64536	-79.37306	Loblaws	4adb6a84f964a520332721e3	43.645427	-79.369789	Grocery Store	132	Highly Recommended
Christie	43.66869	-79.42071	Fiesta Farms	4adcfd7cf964a5203e6321e3	43.668471	-79.420485	Grocery Store	89	Highly Recommended

Using K-Means Clustering Approach:



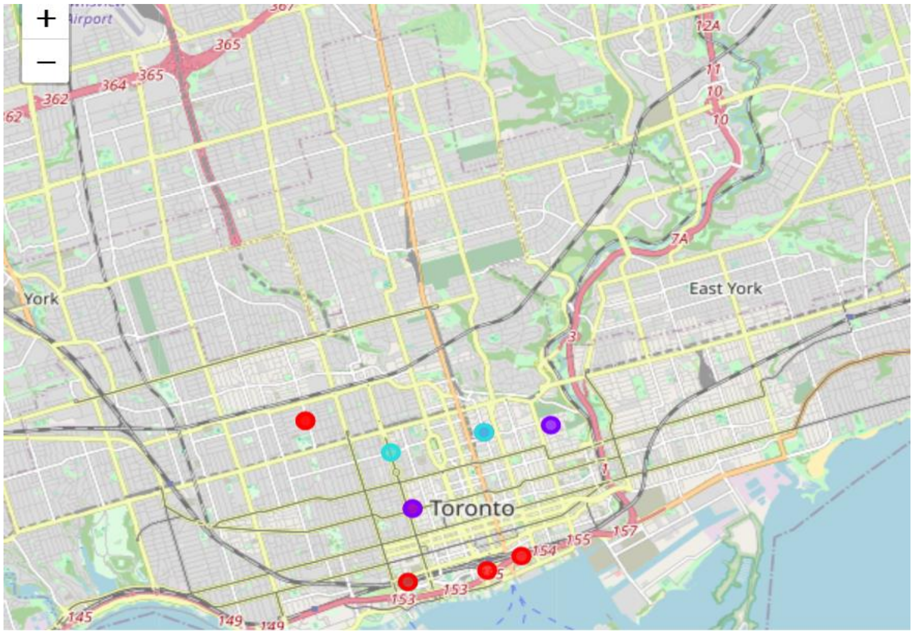
```
# set number of clusters
kclusters = 4

# add neighborhood column back to dataframe
dt_onehot['Neighborhood'] = dt_venues1['Neighborhood']

dt_clustering = dt_onehot.drop('Neighborhood', 1)

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

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

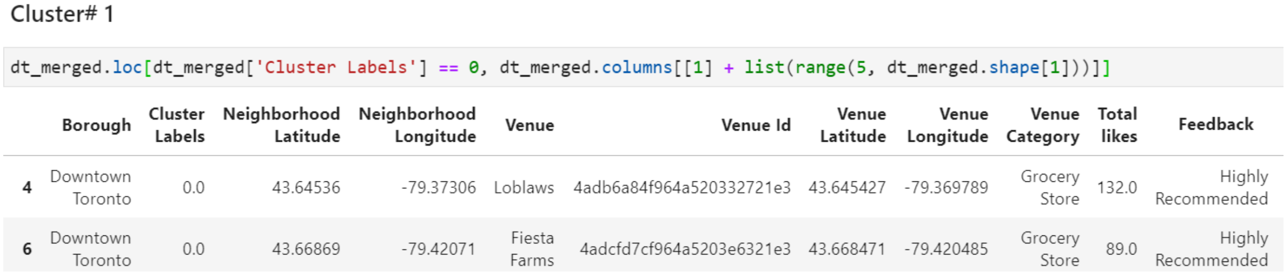
```
array([3, 3, 0, 0, 2, 0, 1, 2, 3, 1], dtype=int32)
```



```
# add clustering labels
dt_venues.insert(0, 'Cluster Labels', kmeans.labels_)

dt_merged = dt_data


dt_merged = dt_merged.join(dt_venues.set_index('Neighborhood'), on='Neighborhood')
dt_merged=dt_merged.dropna()
dt_merged.head() # check the last columns!
```

Visualization of the created clusters.

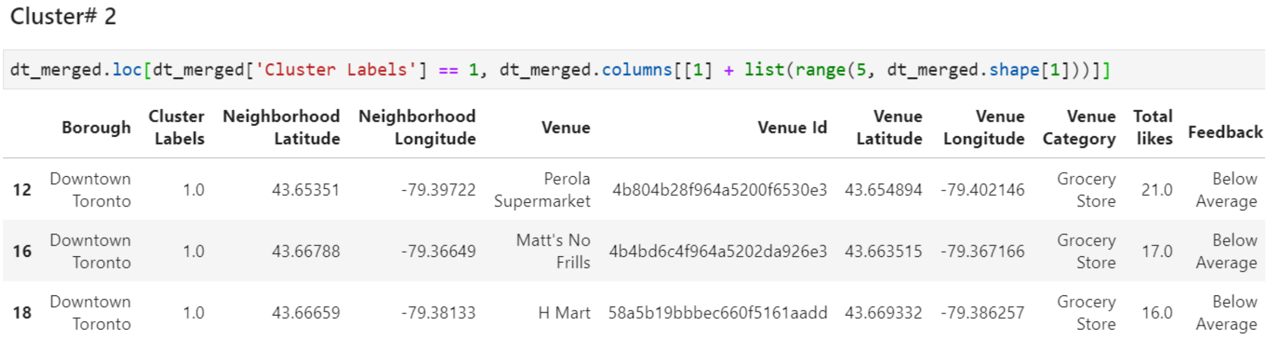




Cluster# 1

```
dt_merged.loc[dt_merged['Cluster Labels'] == 0, dt_merged.columns[[1] + list(range(5, dt_merged.shape[1]))]]
```

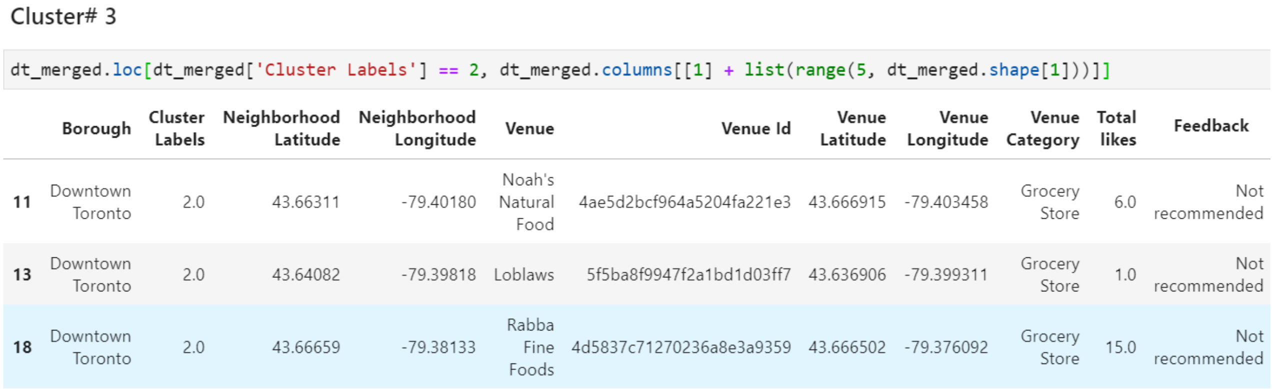
	Borough	Cluster Labels	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Id	Venue Latitude	Venue Longitude	Venue Category	Total likes	Feedback
4	Downtown Toronto	0.0	43.64536	-79.37306	Loblaws	4adb6a84f964a520332721e3	43.645427	-79.369789	Grocery Store	132.0	Highly Recommended
6	Downtown Toronto	0.0	43.66869	-79.42071	Fiesta Farms	4adcfd7cf964a5203e6321e3	43.668471	-79.420485	Grocery Store	89.0	Highly Recommended



Cluster# 2

```
dt_merged.loc[dt_merged['Cluster Labels'] == 1, dt_merged.columns[[1] + list(range(5, dt_merged.shape[1]))]]
```

	Borough	Cluster Labels	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Id	Venue Latitude	Venue Longitude	Venue Category	Total likes	Feedback
12	Downtown Toronto	1.0	43.65351	-79.39722	Perola Supermarket	4b804b28f964a5200f6530e3	43.654894	-79.402146	Grocery Store	21.0	Below Average
16	Downtown Toronto	1.0	43.66788	-79.36649	Matt's No Frills	4b4bd6c4f964a5202da926e3	43.663515	-79.367166	Grocery Store	17.0	Below Average
18	Downtown Toronto	1.0	43.66659	-79.38133	H Mart	58a5b19bbbec660f5161aadd	43.669332	-79.386257	Grocery Store	16.0	Below Average



Cluster# 3

```
dt_merged.loc[dt_merged['Cluster Labels'] == 2, dt_merged.columns[[1] + list(range(5, dt_merged.shape[1]))]]
```

	Borough	Cluster Labels	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Id	Venue Latitude	Venue Longitude	Venue Category	Total likes	Feedback
11	Downtown Toronto	2.0	43.66311	-79.40180	Noah's Natural Food	4ae5d2bcf964a5204fa221e3	43.666915	-79.403458	Grocery Store	6.0	Not recommended
13	Downtown Toronto	2.0	43.64082	-79.39818	Loblaws	5f5ba8f9947f2a1bd1d03ff7	43.636906	-79.399311	Grocery Store	1.0	Not recommended
18	Downtown Toronto	2.0	43.66659	-79.38133	Rabba Fine Foods	4d5837c71270236a8e3a9359	43.666502	-79.376092	Grocery Store	15.0	Not recommended

Discussion:

So, we made 3 clusters on the feedback of Grocery Stores & user can select the best Grocery store in Downtown Toronto based on the feedback. The results would have been much better if 4 clusters were made.

Conclusion:

Overall a decent application of the key concepts of data science & machine learning after completing this Capstone Project. There are areas of further improvement since learning is an evolving process.