Location Recommender System for Future Taxi Service

Piotr Mechlinski

Introduction: Problem Description

A future taxi provider based on autonomous car fleet operating in Downtown of Toronto is looking for a best place for parking and charging his fleet . Customers are ordering the taxi service mainly from Restaurants, Cafes, Breweries and Groceries. The fleet operator wants to choose parking location to optimize time of the pick-up, especially for the best rated spots. This will optimize the perceived level of quality and should be competitive advantage for the company.

The operator should build the parking the closest to its customers to minimize the costs of opeartion. Finding the right location is the ultimate goal of proposed algorithm.

Introduction: Data Needed

1- We will utilize geolocation data on specific borough and the surrounding neighborhoods (latitude and longitude numbers). We will limit our search to the Downtown of Toronto. The Postal Codes that are into that borough would also be needed.

2- We will use the information about venues in different areas of Downtown and to gain that information we will use "Foursquare" location service (basic and advanced information about that venue such as category and popularity average price of the services).

Main Article

Part 1: Identifying Neighborhoods inside "Downtown, Toronto"

- We will utilize postal codes of different areas inside Downtown to find the list of neighborhoods. We will get the required list of codes from

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

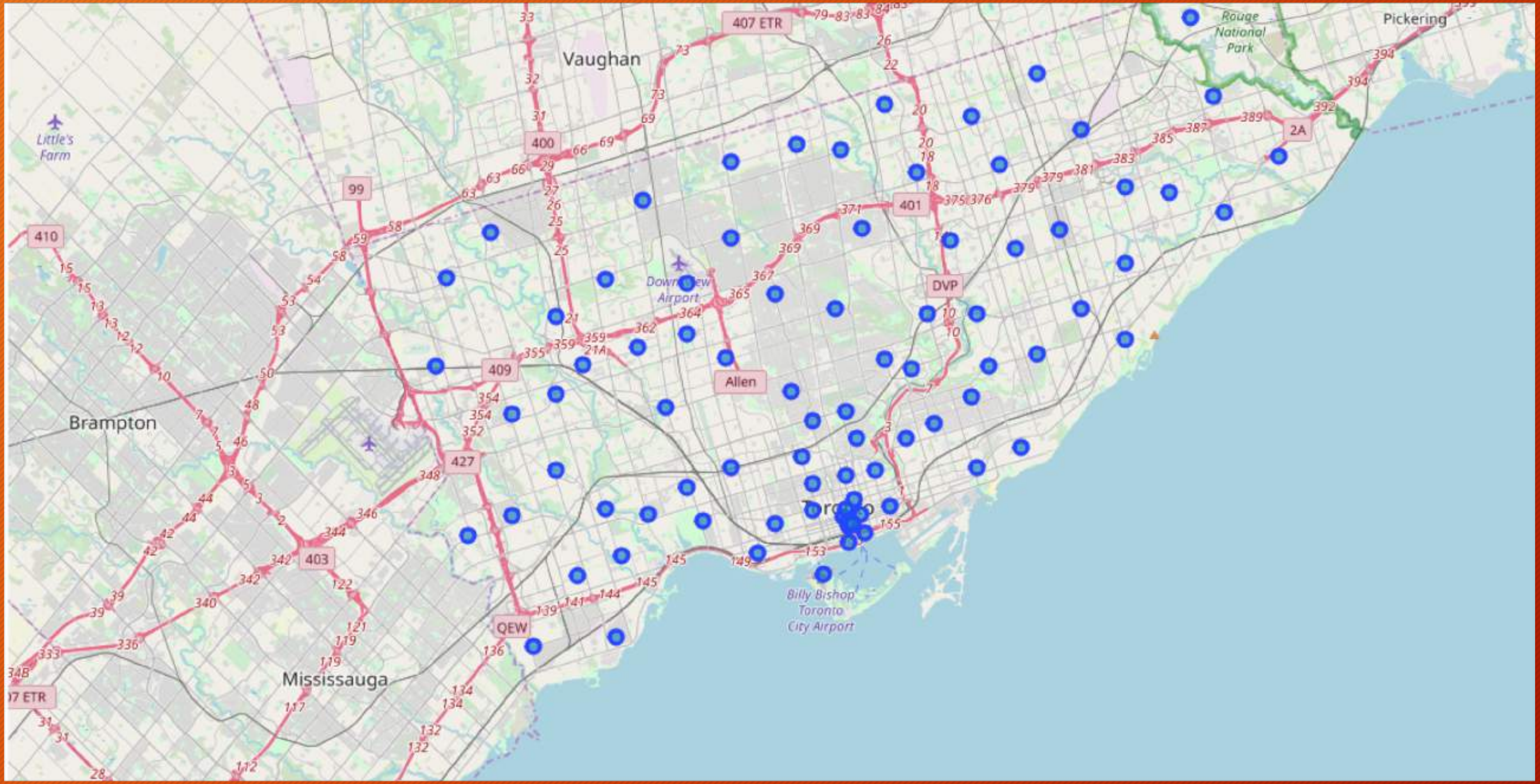
Post Code Data with Coordinates

	Postal Code	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Summary	Venue Category	Distance
0	M4X	Cabbagetown, St. James Town	43.667967	-79.367675	Cranberries	This spot is popular	Diner	140
1	M4X	Cabbagetown, St. James Town	43.667967	-79.367675	F'Amelia	This spot is popular	Italian Restaurant	89
2	M4X	Cabbagetown, St. James Town	43.667967	-79.367675	Butter Chicken Factory	This spot is popular	Indian Restaurant	157
3	M4X	Cabbagetown, St. James Town	43.667967	-79.367675	Kingyo Toronto	This spot is popular	Japanese Restaurant	238
4	M4X	Cabbagetown, St. James Town	43.667967	-79.367675	Merryberry Cafe + Bistro	This spot is popular	Café	173

Part 2: Foursquare Location Data for Venues in Neighborhoods

- After finding the list of neighborhoods, we will connect to the Foursquare to gather information about venues with a chosen distance like 1 km from the center (measured by latitude and longitude of, not the walking distance for venues.)

Part 2: Foursquare data on the map



Part 3: Processing the Data for venues inside the Downtown

- After the data will be gathered, we will perform processing on that data to find needed features for each venue. The main feature is the ventue category which then will be one-hot-coded. After that dataset will be fully ready for machine learning purposes.

Part 3: Focusing on Downtown Area

Part 4: K-Means Clustering

- We will cluster neighborhoods using k-means clustering. We think that 4 clusters is enough for this project. After clustering we will update our dataset and create a column representing the group for each neighborhood.

Part 4: K-Means clustering

```
In [37]: #import k-means from clustering stage
         from sklearn.cluster import KMeans

         # run k-means clustering
         kmeans = KMeans(n_clusters = 4, random_state = 0).fit(final_onehot)
```

```
In [54]: means_df = pd.DataFrame(kmeans.cluster_centers_)
         means_df.columns = final_onehot.columns
         means_df.index = ['G1','G2','G3','G4']
         means_df['Total Sum'] = means_df.sum(axis = 1)
         means_df.sort_values(axis = 0, by = ['Total Sum'], ascending=False)
```

Out[54]:

	Bakery	Breakfast Spot	Diner	Fish Market	Food & Drink Shop	Fruit & Vegetable Store	Grocery Store	Noodle House	Pizza Place	Sandwich Place	Total Restaurants	Total Joints	Total Sum
G3	2.000000	1.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.0	1.000000	2.000000	23.000000	1.000000	32.000000
G4	1.333333	0.000000	0.000000	0.000000	0.000000	0.000000	0.666667	1.0	1.666667	1.000000	11.333333	2.000000	19.000000
G2	2.333333	0.666667	0.000000	0.333333	0.000000	0.000000	1.666667	0.0	1.333333	0.333333	7.666667	1.333333	15.666667
G1	0.333333	0.222222	0.222222	0.000000	0.111111	0.111111	0.333333	0.0	1.000000	0.555556	2.777778	0.666667	6.333333

Reporting Results

- In this part we will 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.

Final result – Parking location

```
neigh_summary
```

Out[55]:

	Neighborhood	Group
0	Agincourt	3
1	Agincourt North, Milliken	4
2	Birch Cliff	1
3	Cedarbrae	2
4	Clairlea, Golden Mile, Oakridge	1
5	Cliffcrest, Cliffside	1
6	Dorset Park, Scarborough Town Centre, Wexford ...	4
7	Highland Creek, Rouge Hill, Port Union	1
8	Ionview, Kennedy Park	1
9	Maryvale, Wexford	2
10		
11		
12		
13		
14		
15		

In [58]:
```
neigh_summary[neigh_summary['Group'] == 1]
```

Out[58]:

	Neighborhood	Group
2	Birch Cliff	1
4	Clairlea, Golden Mile, Oakridge	1
5	Cliffcrest, Cliffside	1
7	Highland Creek, Rouge Hill, Port Union	1
8	Ionview, Kennedy Park	1
10	Morningside, West Hill	1
11	Rouge, Malvern	1
12	Scarborough Village	1
15	Woburn	1

In [64]:
```
name_of_neigh = list(neigh_summary[neigh_summary['Group'] == 4]['Neighborhood'])[0]
scarborough_venues[scarborough_venues['Neighborhood'] == name_of_neigh].iloc[0,1:5].to_dict()
```

Out[64]:
```
{'Neighborhood': 'Agincourt North, Milliken',
 'Neighborhood Latitude': 43.815252200000003,
 'Neighborhood Longitude': -79.284577200000001,
 'Postal Code': 'M1V'}
```