

Analyzing Toronto and Manhattan Similarity

Coursera | IBM Data Science | Capstone





I. Introduction

- Small business seeks to expand from Toronto to America:
 - Select city based on similarity of venues around each neighborhood -- Manhattan is potential choice.
 - Find **similarity** between Toronto and Manhattan.
 - Find most **common venue types** between cities.
 - Allows better client targeting and consistent business model between cities.



II. Data Acquisition

- Canadian postal codes from [Wikipedia](#), coordinate data from [Coursera](#), New York from [Coursera](#), location data from [Foursquare API](#).
- Processed data frame w/ borough, neighborhood, latitude, longitude + top 5 venue types.
- Neighborhoods in **Toronto = 38, Manhattan = 40**.
- Top 5 venue types for each neighborhood analyzed.



III. Exploratory Data Analysis

City	Neighborhoods	Venues	Venue Type	Top Venues	Top Venue Types
Toronto	38	1689	236	190	81
Manhattan	40	3317	337	200	65

Top Venue Type	Coffee Shop	Cafe	Italian Restaurant	Park	Hotel
Avg. Frequency	0.1029	0.0776	0.0714	0.0437	0.0333

Cluster	0	1	2	3	4
Neighborhoods	13	4	59	1	1

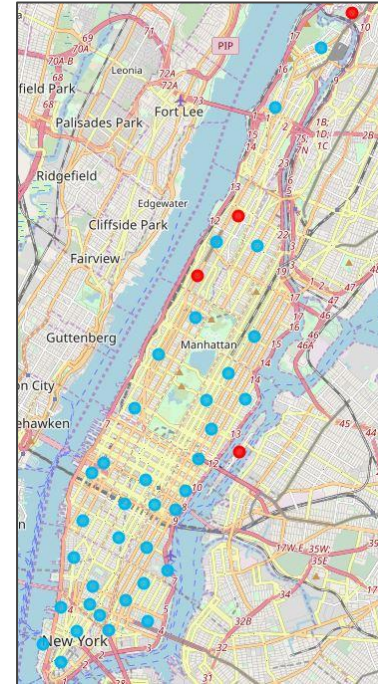
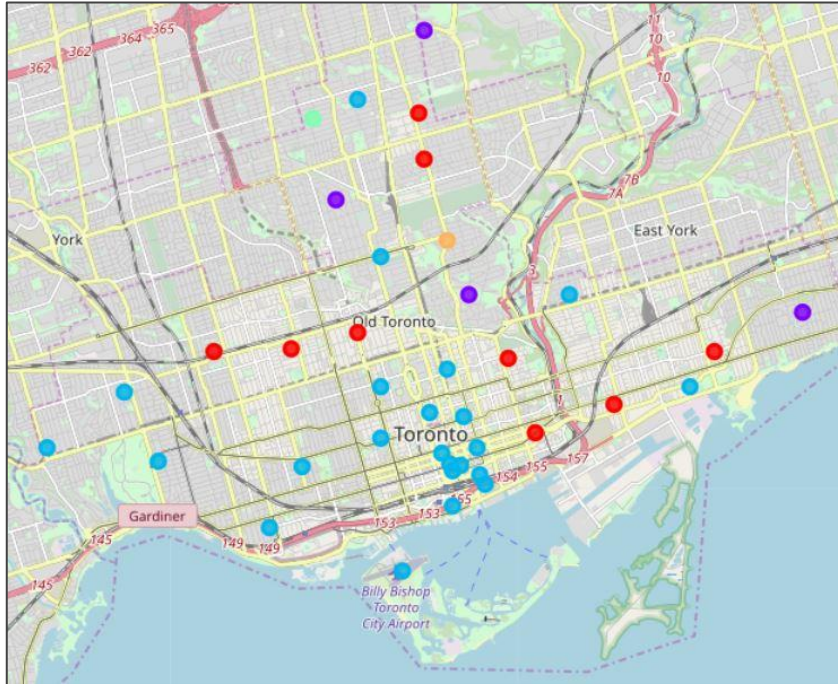


IV. Results

City	Venues/Neighborhood	Venue Similarity	Similarity Index
Toronto	44	66%	50.868%
Manhattan	83		

Cluster	0	1	2	3	4
Distribution	16.667%	5.1282%	75.641%	1.2821%	1.2821%
Label	Store	Park/Bus	Food/Dining	Garden	Playground

IV. Results Continued





V. Discussion

- Food/dining venue types overrepresented (Cluster 2) -- **75% of neighborhoods, top 3 venue types represent 25% of top venues.**
- Concentrated in **Central Toronto, uniformly in Manhattan.**
- Focus specifically on **coffee shops/cafes.**
- **Recommend Manhattan as potential choice (similarity>50%).**



VI. Conclusion

- Demonstrates similarity between Toronto and Manhattan.
- Potential process refinement and direction:
 - **Direct integration of geographical data** (no postal codes).
 - **Optimize location data** -- maximize coverage and minimize overlap/redundancy.
 - **Apply to multiple cities** to find optimal city/neighborhood.