# Coffee Crawl in Toronto

By Morgan Bounds

# 1. Introduction

## 1.1. Background

For the coffee enthusiast, one of the most exciting aspects of traveling is exploring new coffee shops. A popular method for such exploration is known as the “coffee crawl”. A coffee crawl consists of touring multiple coffee shops in a given location, so that one may sample coffees from various establishments in a single day. This approach is particularly well suited to those operating under time constraints, as one can maximize their coffee intake variety over a relatively short period of time.

## 1.2. Problem

Assume someone is going on a business trip to Toronto. They will be in and out of meetings for most of the trip, but they will have six hours of free time to explore the city. They would like to use this time to go on a coffee crawl. Ideally, they would like to spend at least twenty minutes in each coffee shop they visit, and they would like to visit as many coffee shops as possible. They will not have a car, so they must be able to walk to each coffee shop. Furthermore, they must start and end at their hotel. The person’s company will allow them to choose whichever hotel in Toronto they want to stay at, so which hotel should they choose to optimize their coffee crawling experience?

# 2. Data Acquisition and Cleaning

## 2.1. Data Sources

Postal code data for different Neighborhoods in Toronto was gathered from Wikipedia (Canadian Postal Codes). Then, geospatial data was gathered by utilizing a file provided by Coursera (Canadian Geospatial Data). Finally, venue data for hotels and coffee shops was gathered using the Foursquare API (https://api.foursquare.com/).

## 2.2. Data Cleaning

The postal code data scraped from Wikipedia contained some rows where the Borough/Neighborhood was "Not Assigned". We immediately dropped all such rows. Then, we utilized the geospatial data from Coursera to add latitude and longitude columns to our data frame, one for each Borough/Neighborhood row.

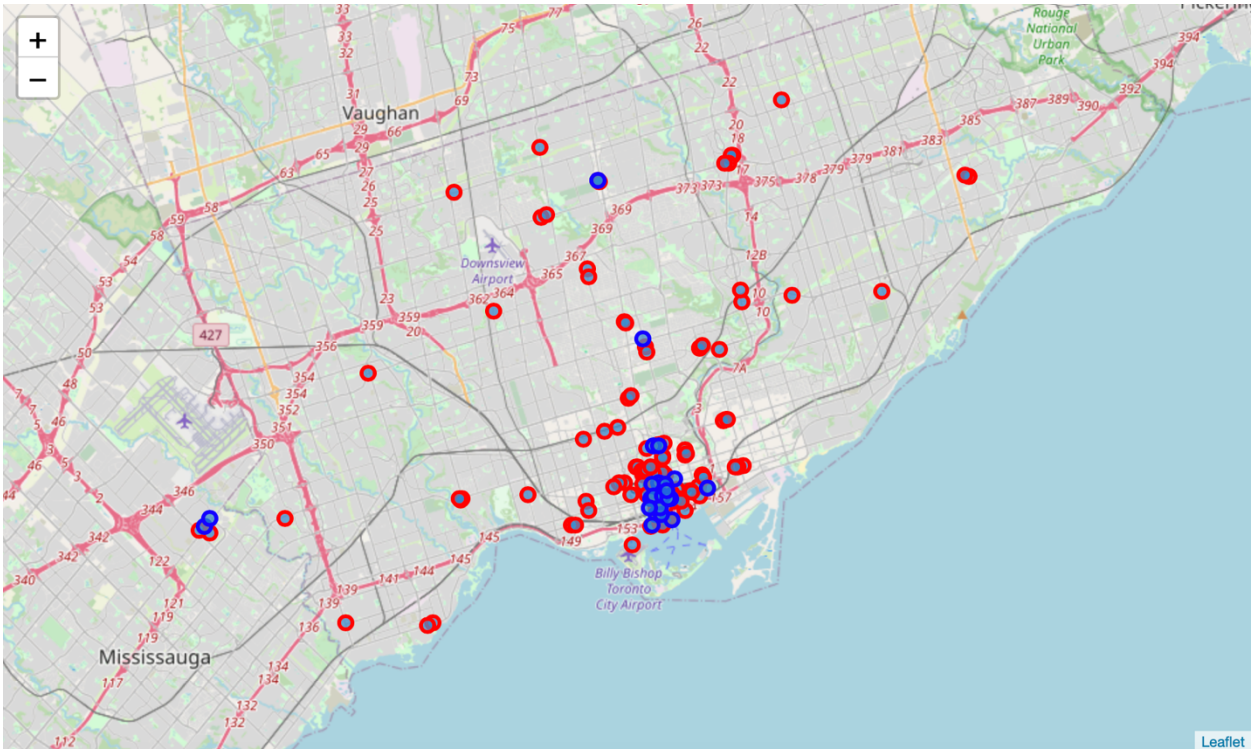
Next, we split the venue data from Foursquare into two separate data frames: one which only contained coffee shop venues and one which only contained hotel venues. Some venues bordered between several neighborhoods, and so there were multiple entries of identical venues under different neighborhood names. We removed identical venues by making sure that there was only one row for each unique pair of latitude and longitude coordinates. This left us with data for 135 unique coffee shops and 24 unique hotels.

## 2.3. Feature Selection

For this problem, the main question we seek to address involves the locations of specific venues. Thus, the only features we will be using are the venue names and their corresponding latitudes and longitudes.

# 3. Exploratory Data Analysis

Utilizing Folium, a popular Python library for geospatial data, we created a map of our venues. Coffee shops are represented by red dots and hotels are represented by blue dots.




Clearly, the downtown areas of Toronto (lower center region by the water) contain the most hotels and coffee shops. It immediately becomes clear that getting a hotel downtown will probably be our business traveler's best bet.

# 4. Cluster Modeling

## 4.1. Model Selection

Since our problem involves someone taking a walking tour of numerous coffee shops, a clustering algorithm seemed only natural. While K-Means Clusters is a popular machine learning algorithm, it does not necessarily contain any information on the density of the clusters it labels. Ideally, we want to find a dense cluster of tightly packed coffee shops, so it would be easy for

someone to quickly tour them all on foot. Thus, we decided to implement the DBSCAN (Density Based Spatial Clusters for Applications with Noise) algorithm.

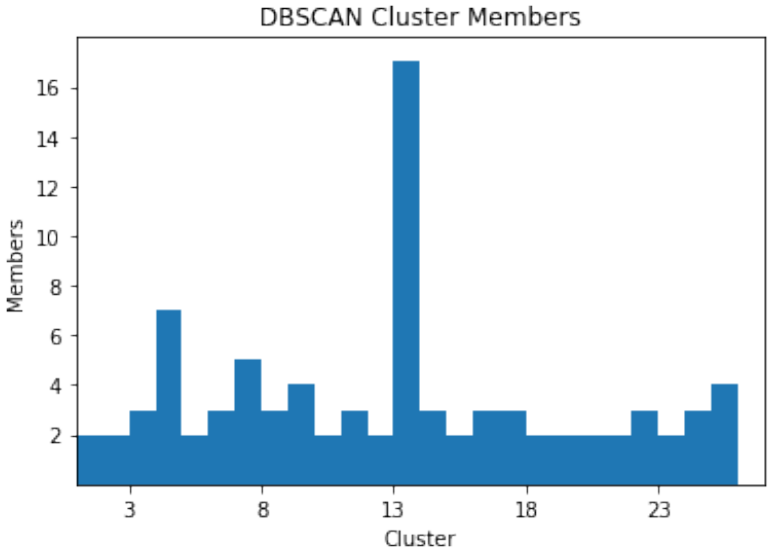
The DBSCAN algorithm works by labeling clusters based on a point's proximity to other points. One of its parameters is a radius value. Points are basically assigned to the same cluster if one can "connect the dots" between the two points in such a way that each "connection" is less than the radius value. In our problem, this radius represents the maximum distance someone would have to walk between coffee shops during their coffee crawl. Thus, ideally this radius is minimized while still having a reasonably high number of coffee shops in a cluster.

In order to perform the DBSCAN algorithm, we needed to create a distance matrix which contained the distance from each coffee shop to each other coffee shop. Since the earth's surface is roughly spherical, we decided to compute the geodesic distance instead of the Euclidean distance. We computed these values using the latitude and longitude coordinates of each point, and then essentially finding the arc length by leveraging the earth's approximate radius.

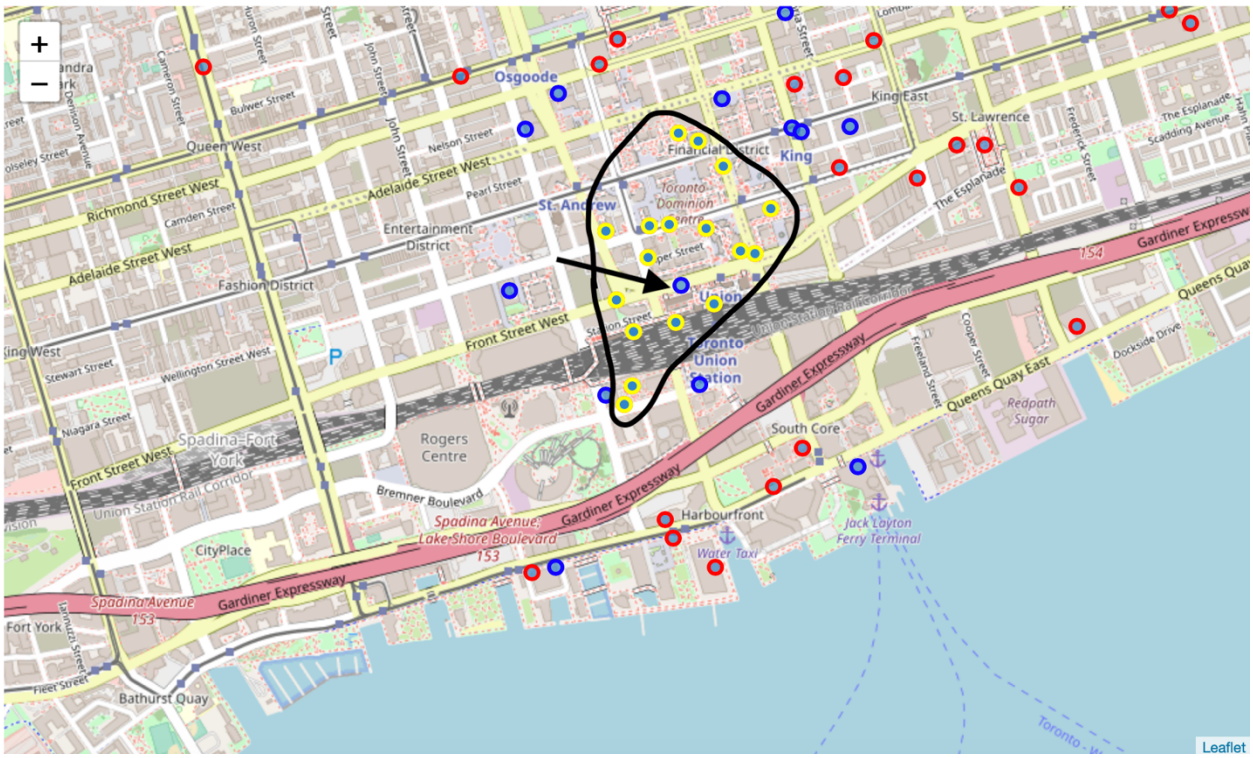
## 4.2. Model Execution

After trying several parameters for the DBSCAN algorithm, we had the most success with a radius value of approximately .17 kilometers and a minimum number of 2 samples per cluster. Note that the average human walking speed is 5 km/hour (https://en.wikipedia.org/wiki/Walking), so a radius of .17 km means that one could traverse a given cluster by taking roughly two minute walks in between points. The model identified 26

clusters, and their number of members is represented in the following histogram.




Clearly, Cluster 13 had the most members with 17 total coffee shops assigned to this cluster. The next biggest cluster falls much shorter with only 7 total members. To help visualize Cluster 13, we add it to our previous map by changing the color of coffee shops which belong to Cluster 13 from red to yellow.



# 5. Conclusions

The algorithm has successfully identified a densely packed cluster of 17 unique coffee shops. Furthermore, from looking at the map, there is clearly a single hotel right in the middle of the cluster where the arrow is pointing. This hotel is “The Fairmont Royal York”, and evidently this is our business traveler’s best bet for their coffee crawl! With 17 coffee shops, spending 20 minutes at each one would take roughly 5 hours and 40 minutes. With roughly 2 minutes in between each coffee shop, this adds up just a little over six hours, which was the goal. Therefore, they should be able to spend most of the six hours experiencing different coffee shops while only spending a minimal time traveling in between shops.

# 6. Future Directions

In the future, we could make the solution more robust by considering more features than just location. For instance, we could also use the Foursquare API to get ratings and reviews for each coffee shop, and then try to maximize the overall rating of the coffee shops visited during the coffee crawl. We could also consider different starting points. Rather than choosing an ideal starting point (like the ideal hotel) we could try to identify key coffee crawl tours relative to any given starting place in Toronto.