Hotel Recommendations Based on Popular Nearby Venues

Introduction

Background

Traveling can always incite stress, whether it is planning for the trip or while on the trip. One area people put a lot of thought into is what hotel to stay at when traveling to a new location. Whether traveling for business or pleasure, people may know they want to stay near a particular venue. Often, there are several hotels located nearby. Cost may be the primary reason people choose a hotel, but if there are several hotels in a particular location with similar prices, how does one choose so that they are likely happy where they stay?

Everyone has different tastes and preferences. Often, the extent to which someone is happy where they stay is impacted by the surrounding area. Some people may prefer hotels with a variety of restaurants and shopping. Others may prefer a hotel that is near places with sights to see and things to do. Still, others may be traveling for short work trips and only need a few places to eat nearby.

Problem and Audience

The problem is that when searching for a place to stay, travelers can get good information on a hotel's price, ratings, and ammenities but know very little about the area surrounding the hotel. One of the worst experiences for a traveler is arriving at a hotel to discover there are no restaurants in walking distance and the traveler does not have a car! Travelers can look up each hotel they are interested in through Google and look at what's nearby. However, if a traveler has multiple hotels to choose from in different areas, it can be difficult to remember and manually track the nearby venues and which areas might be the best.

The purpose of this project is to pilot test the feasibility of using different data sources to identify groups of hotels based on the popular venues located nearby for would-be travelers, the intended audience. This project focuses on hotels in Toronto, Canada.

Data

Data Sources

To provide detailed hotel information to help travelers figure out where to stay in Toronto, Canada, the data used for this project are from Hotels.com and Foursquare. To provide travelers with an estimate of hotel prices, I scraped Hotels.com using the Selenium package in Python. I searched for hotels in Toronto for one night (Friday to Saturday) with two adults in mid-November 2020 (to ensure few were sold out) and sorted the results by distance from city center. The three pieces of information scraped from Hotels.com were hotel name, hotel address, and hotel price (see Table 1 for a summary of information used from each data source).

There were several different types of Foursquare data pulled using the API. One dataset used the first two words of the hotel names from the Hotels.com data to search for these specific hotels in the Foursquare data. This Foursquare data provided the venue ID for each hotel in order to obtain venue details for each hotel. The combined data from Hotels.com and Foursquare provided hotel name and address, price information, ratings, latitude and longitude, and contact information. The other data from Foursquare consisted of venues within a 500-meter radius around each hotel.

Table 1. Data Sources and Information Used for Pilot Test

Hotels.com	Foursquare Hotel Information	Foursquare Nearby Venues
Hotel name	Hotel venue ID	Venue Name
Hotel address	Hotel latitude	Venue Category
Hotel price	Hotel longitude	
	Contact information	
	URL	
	Ratings	
	Likes	

Data Cleaning

Before the different data sources could be merged together, each dataset had to be cleaned. The following sections describe how each dataset was cleaned.

Hotels.com Data

After searching for the parameters described in Data Sources above, the resulting URL was used for scraping. I used the Selenium package from Python to scrape the Hotels.com data. One issue with this particular website was that it didn't provide the results in pages but used infinite scrolling. If the code did not account for the infinite scrolling, it only scraped the first 10 results. After taking into account the infinite scrolling, the script scraped 138 hotels in the Toronto area.

There were only a couple of steps needed to clean this data. An initial look at the data revealed there were two hotels with no prices. After looking at the search page from Hotels.com, I noticed that two hotels were sold out. These two hotels were removed from the data. In the data, the price data type was text instead of numeric. This was because the scraped data

included words and numbers because the scraping captured words like "Only 3 left at this price." Because the price was always at the end of the string, I used regular expressions to extract the numbers at the end of the string that represented the price. I then created a clean dataset after dropping the original price variable and renaming the new price variable that was now numeric.

Foursquare Data

There were three API calls to Foursquare for different types of data. The first call to Foursquare was used to add basic details to the 138 hotels in Toronto, Canada identified through Hotels.com. This was also the best way to get the hotel's venue ID in Foursquare which I needed for premium calls to get more specific hotel details. The API call to Foursquare searched the Toronto area using a search of the first two words of each hotel name. Next, I examined the size of the Foursquare dataset overall and the size of the dataset when only focusing on those with the hotel category.

Next, I created a new dataframe with only those identified as hotels. This dataset was reduced to include the columns: "id," "name," "category name," "location.address," "location.lat," and "location.lng." The remaining variables were dropped from the dataset either because that had mostly NaN values or were not relevant to goal of this project.

The next step was to merge this basic hotel data from Foursquare with the hotel data from Hotels.com. I initially tried to match the Foursquare data and the Hotels.com data using the full name. However, a quick review of the names revealed that some varied by a few words within each dataset (Hotels.com and Foursquare), which would make matching by hotel name for a merge difficult. I first tried using the first two words in the name but that still made things difficult for chain hotels in different parts of the city. In the end, the best way to merge the two datasets was through the address.

In order to match and merge by street addresses, I had to transform the Hotels.com dataset by splitting the address into multiple columns. The address variable in the Hotels.com dataset included the full address with key pieces separated by commas. I used the comma to separate each parts of the address into separate columns. This processes resulted in a column with only the street address which matched the format of the street address in the "location.address" in the Foursquare data.

In breaking apart the address variable in the Hotels.com data, I had to test it first before saving it as a dataframe. This is because I kept getting errors, and after testing out the str.split code, I realized that some address entries had additional information after Country. Since the purpose was to isolate the street address for merging, I ensured the new dataframe split into the correct number of columns and then removed the extra columns from the split.

I examined the Foursquare data to minimize any issues with merging the two datasets. I realized there were some duplicates that could cause issues with the merge. I also noticed some addresses had NaN values. After removing duplicates and rows with NaN values in "location.address," I merged the Hotels.com data and the Foursquare data.

The printed shape of the merged data revealed that there was only a specific match for 46 of the 138 hotels in the Hotels.com data. I used code to identify which hotels in the Hotels.com data were not part of the merged data. After comparing the addresses between those in the Hotels.com data with the Foursquare data, it was clear that some didn't merge because of differences in whether or not the address had abbreviations like E for East or Dr for Drive.

One way to capture the remaining hotels for the merge was to use the number from the street address and the first word of the street which was usually was not abbreviated. Similar to what I did earlier, I split the Street variable in both datasets and used the first two columns that resulted from the split containing the street number and first word of the street.

From a visual inspection of the data, there appeared to be several duplicates in the merged data which raised the question of whether there were duplicates in the Hotels.com data. After checking the data, there were no duplicates in the Hotels.com data and the duplicates may have been an artifact of the merge. The merged data was cleaned to drop the duplicates based on the Address variable. This resulted in a merged dataset with 114 hotels.

After printing the hotels that did not have a match in the Foursquare data, I searched some of their names in the Foursquare data. There were four I found in the Foursquare data that did not merge due to minor differences in the address. The remaining hotels did not match and were dropped from the analysis. After adding the additional four hotels with a match in the Foursquare data, I cleaned the merged hotels data to remove columns I wasn't going to use (e.g., columns based on the split address).

The next dataset I created from Foursquare involved getting the details for each hotel. The detailed information from Foursquare allowed me to provide travelers with the average venue rating and total number of likes as additional pieces of information to make an informed decision. The detailed information also included the venue URL and contact phone. The second API call to Foursquare resulted in 130 columns of data. Because I knew I wasn’t going to use all of the data in the columns, I identified which columns of information I was interested in using and cleaned the data so that the resulting dataset had the hotel name, id, contact information, hours information, ratings, and likes. This data was merged with the previously merged hotel dataset using the Foursquare id associated with the hotels.

The new merged dataset needed a little cleaning before the last API call to Foursquare. The new merged data resulted in 4 additional rows than expected. Upon further inspection, there were two address with several duplicates. Some of these duplicates had varying hotel names between Hotels.com and Foursquare. For the first set of address duplicates, I dropped all but the first one (Hampton Inn by Hilton Toronto Airport). For the second set of duplicates, I dropped all 4 because there was no additional detailed Foursquare information (e.g., all values were NaN). Using the index numbers for these duplicates, I deleted 7 rows from the dataset.

The last API call to Foursquare was to obtain the venues near each hotel within a 500 meter radius. Because venue category included 'Hotel,' those noted as Hotel were removed from the venue dataset. The data were then transformed to a new dataset that showed the top 10 most common venues for each hotel. To do this transformation, each venue was dummy coded to get

the mean frequency of each venue for each hotel. The mean frequency was then used to identify the top 10 most common venues for each hotel. This dataset was then merged with the last merged dataset by venue ID to create a master dataset of hotel information.

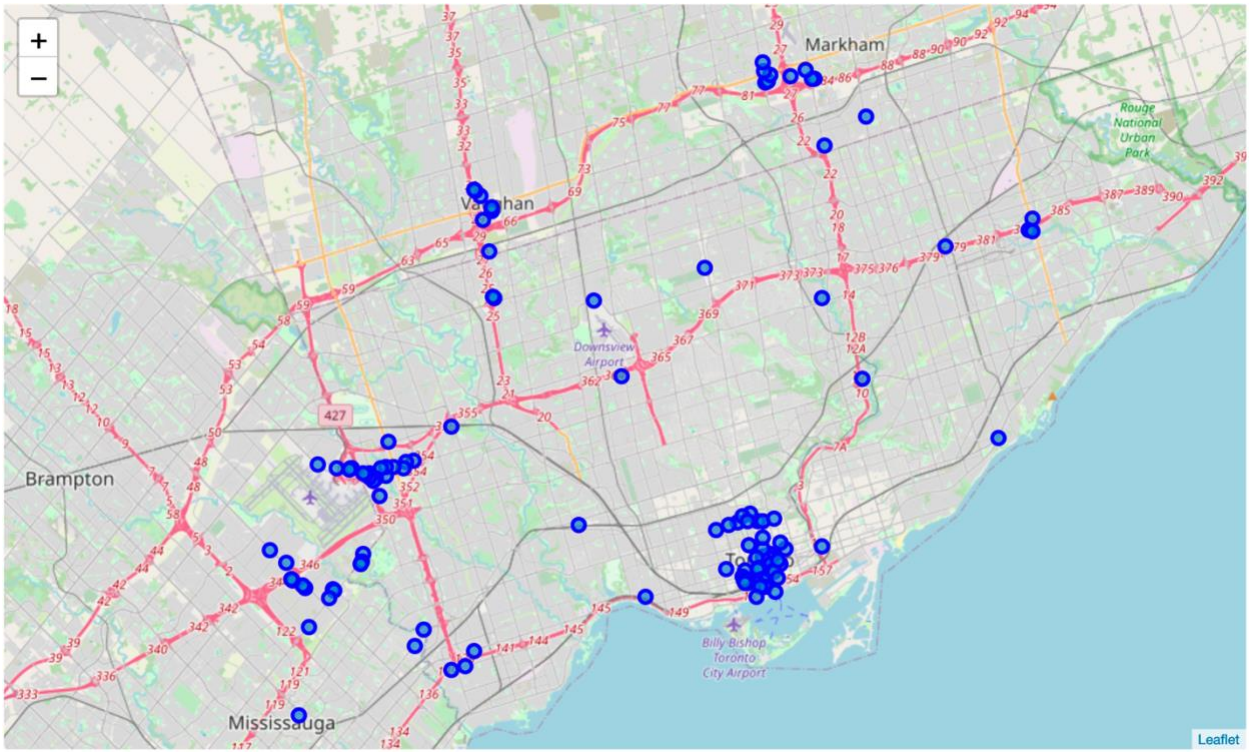
Methodology

This section describes the exploratory analysis and the data analysis of the hotel data. The purpose of the exploratory analysis was to get an idea of where the hotels were located in Toronto, Canada and to get a snapshot of the relationship between price and number of venues near hotels. The data analysis describes the k-means clustering approaches used to group the hotels based on nearby venues.

Exploratory Analysis

First, I mapped the hotels in Toronto, Canada (Figure 1). The purpose of the map was to identify if there were clear groups or clusters of hotels. The theory was that hotels that were located near each other were more likely to have similar venues nearby. Figure 1 reveals that there appear to be several clusters of hotels. Hotels within each cluster may have similar nearby venues and hotels in different clusters may vary in the kinds of venues that are nearby. A clustering analysis based on latitude and longitude would confirm how many hotels were near each other. This is described in the data analysis section.

Figure 1. Map of Hotels in Toronto, Canada



The next part of the exploratory analysis was to examine if there was a relationship between hotel price and the number of venues nearby. For example, is it the case that hotels near a

variety of restaurants and things to do are more expensive because they cater to tourists? In order to get a visual representation of this relationship, I transformed the data to summarize the number of hotels within a given price range and with a given range of nearby venues. Figure 2 is a heatmap of the resulting data.

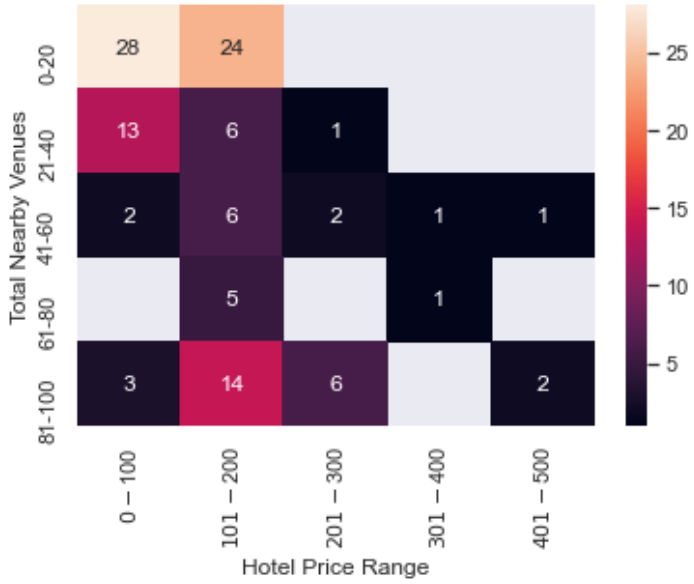


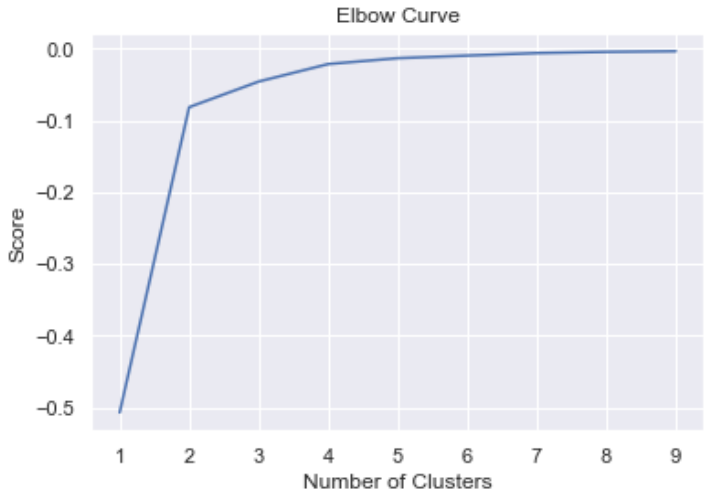
Figure 2. Heatmap of Number of Nearby Venues by Hotel Price Range

The heatmap shows that hotels that are $200 or less are more likely to have less than 20 nearby venues. Hotels greater than $200 were more likely to have 80-100 nearby venues and were never in areas with less than 20 venues. There were some hotels that cost less than $200 a night that were in areas with 21-100 nearby venues. While there was a slight relationship, it was not consistent. This means that travelers cannot use price alone to determine if a hotel will have a variety of venues nearby.

Data Analysis

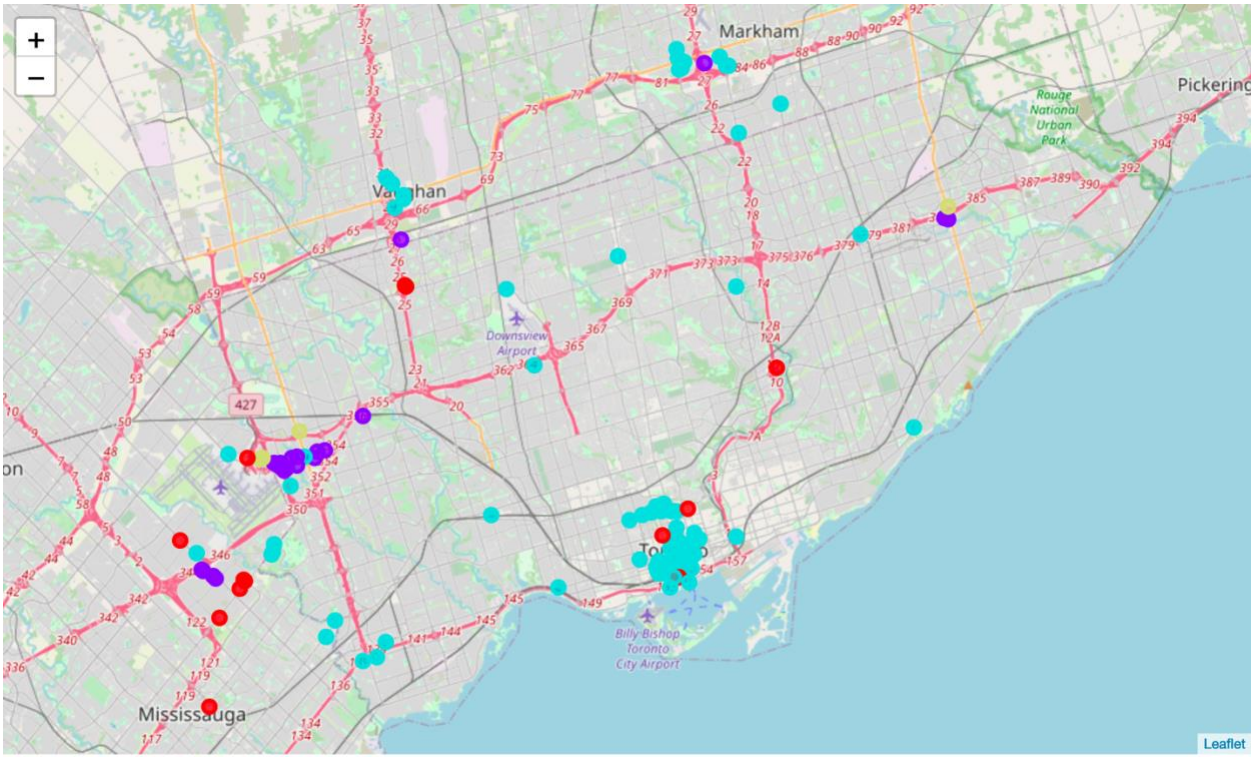
In order to get an idea of how many clusters to look for when clustering hotels by common venue categories, I tested for the best number of clusters based on each hotel's latitude and longitude. My theory was that hotels located near each other would share similar types of venue categories nearby. The elbow curve revealed that the max number of clusters that explained the variance was 4 before leveling off and accounting for minimal variance.

Figure 3. Elbow Curve of K-means Clusters for Hotels Based on Latitude and Longitude



The next step was to conduct a k-means cluster analysis of the average number of venue categories for each hotel. Using four clusters based off the k-means cluster analysis of hotel latitude and longitude, I ran another k-means cluster analysis. Next, I added the resulting cluster labels to the hotel data. Last, I created a new map of the hotels in Toronto that color coded each of the clusters from the cluster analysis (Figure 4).

Figure 4. Map of Hotel Clusters in Toronto Based on Most Common Nearby Venue Categories



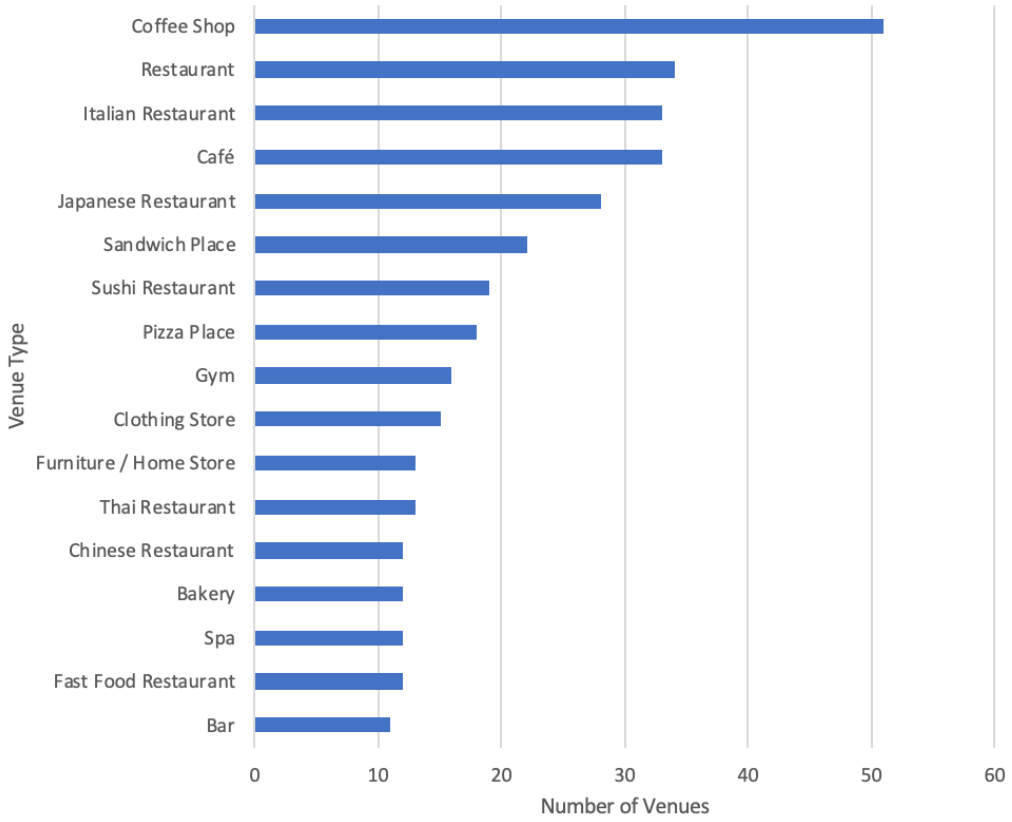
Results

The K-means clustering analysis and visual representation of the cluster analysis (Figure 4) show groups of hotels based on nearby venue categories. The largest cluster appears blue on the map. In order to get an idea of potential differences between clusters, I examined the total number of venue categories or types associated with each cluster.

Blue Cluster

There were 76 hotels within the blue cluster. This cluster contained a wide variety of places to eat and things to do (Figure 5). Common places included coffee shops and cafes, a variety of places to eat, shopping, gyms, spas, and bars. If someone wanted to stay at a hotel with a variety of things to do and place to eat within walking distance, hotels in the blue cluster would be the place to stay.

Figure 5. Total Number of Venue Types for Hotels in the Blue Cluster



In addition, these hotels were not located in one area of Toronto. While many were in downtown Toronto, there were plenty of choices near the airport, Vaughan, Markham, and scattered throughout Toronto. Depending on travelers' reasons to visit Toronto, they would likely find a hotel with a variety of options nearby if choosing from the blue cluster of hotels. Hotel prices in this cluster ranged from $40 to $453 (Table 2). The average hotel price was $142.97. Fifty-percent of the hotels in this cluster were less than $125. The average rating for hotels in this cluster was 7.0 and the average number of likes was 50.

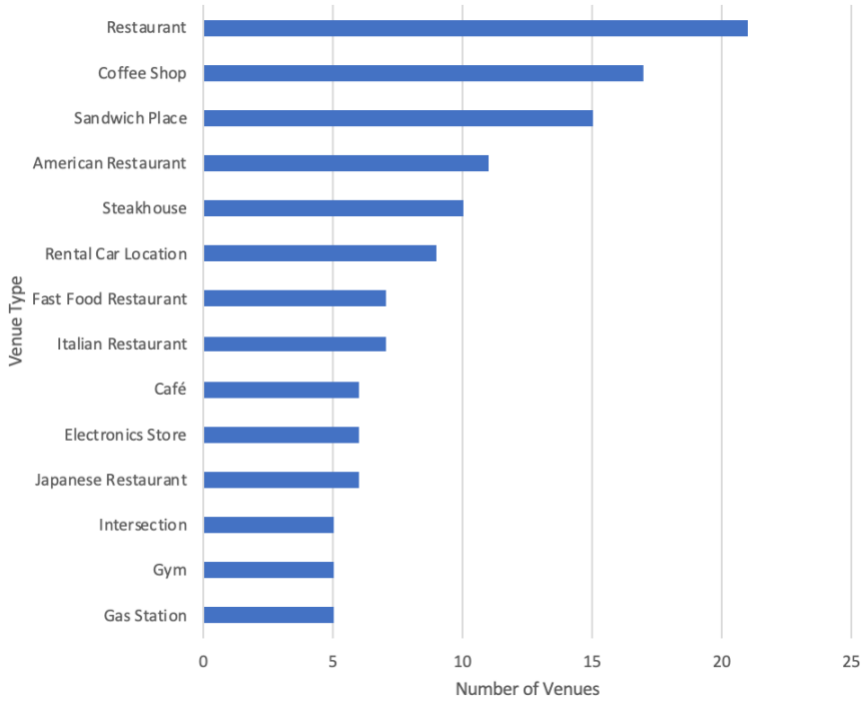
Table 2. Summary Statistics for Hotels in the Blue Cluster on Price, Rating, and Likes Count

	Price	Rating	Likes Count
count	76	63	76
mean	$ 142.97	7.0	50.3
min	$ 40.00	4.8	0
25%	$ 88.00	6.3	7
50%	$ 124.50	7.2	20
75%	$ 158.75	7.9	54.3
max	$ 453.00	9.2	395

Purple Cluster

The purple cluster consisted of 21 hotels. While there was still some variety, there were not as many options as the blue cluster (Figure 6). Travelers who may stay in these locations may be coming to the area for work for example. They may have less time for sight seeing and participating in a variety of activities in the area but still want a couple of different places to eat when they are not working. As such, the purple cluster may be a good choice for them. In addition, this cluster is more likely to have rental car stores and gas stations making it convenient to pick up and drop off rental cars.

Figure 6. Total Number of Venue Types for Hotels in the Blue Cluster



The hotels in this cluster are less expensive than the blue cluster. They ranged in price from $66 to $189 (Table 3). The average price is $105.19 and half of the hotels were less than $106. The average rating for hotels in this cluster was 6.1, and the average number of likes was 16.

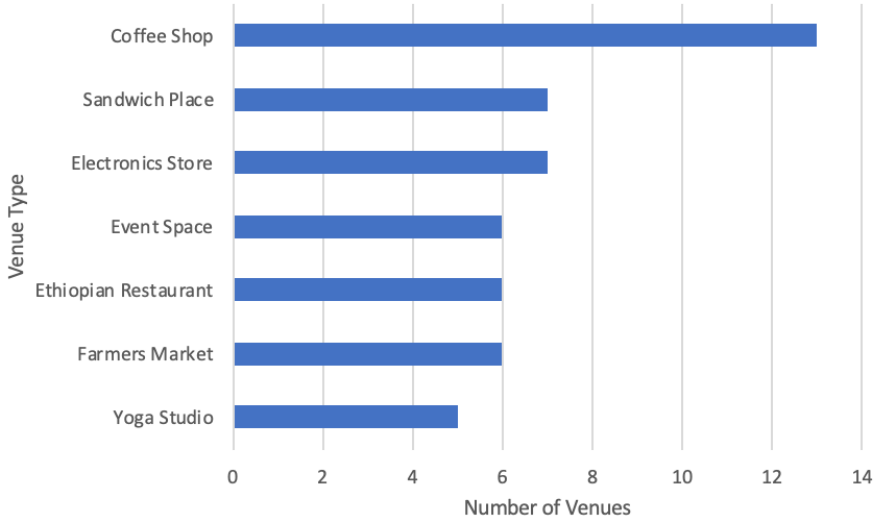
Table 3. Summary Statistics for Hotels in the Purple Cluster on Price, Rating, and Likes Count

	Price	Rating	Likes Count
count	21	16	21
mean	$ 105.19	6.1	16.9
min	$ 66.00	4.8	0
25%	$ 82.00	5.5	5
50%	$ 106.00	6	14
75%	$ 119.00	6.5	24
max	$ 189.00	7.6	44

Red Cluster

There were 13 hotels in the red cluster. The red cluster consisted of only two types of places to eat: sandwich places and Ethiopian restaurants (Figure 7). There were plenty of coffee shops in the red cluster. Those who stay in these hotels may choose these hotels because they are attending something at an event space. However, it would be important for travelers to know before booking their hotels that there are not a lot of places to eat within walking distance and may need to drive to get more variety when eating or if they need to shop. This type of analysis by nearby venues can help them make informed decisions and improve their travel planning.

Figure 7. Total Number of Venue Types for Hotels in the Red Cluster



Given the lack of variety nearby, it was surprising that the hotel price ranges were from $59 to $348 (Table 4). The high price of hotels is likely due to the event spaces near hotels in this cluster. The average price for a hotel in this cluster was $123.46. Half of the hotels in this cluster were $98 or below. The average rating was 6.7, and the average number of likes was 20.

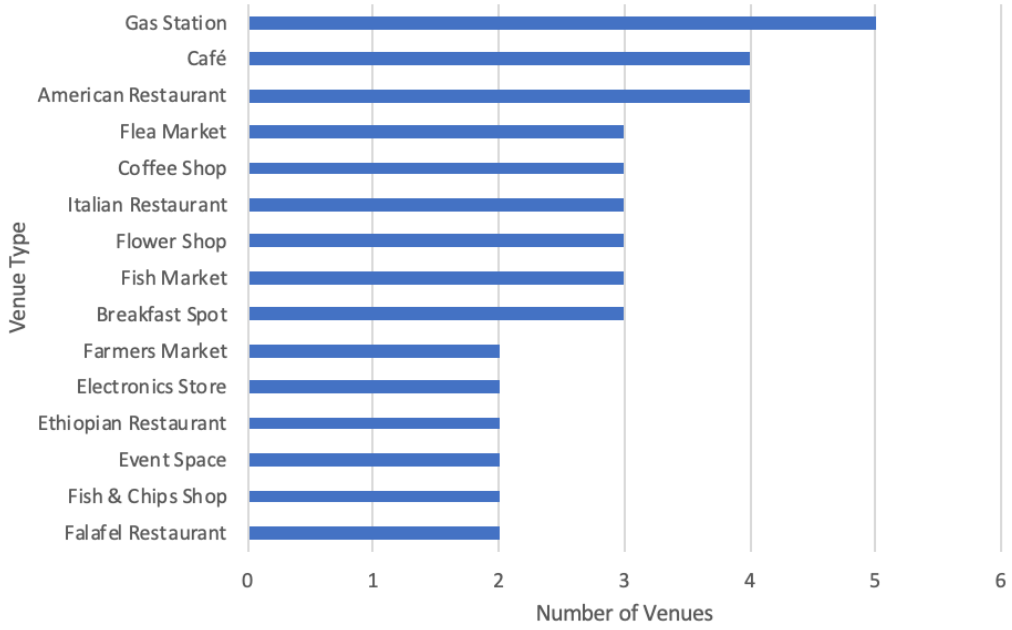
Table 4. Summary Statistics for Hotels in the Red Cluster on Price, Rating, and Likes Count

	Price	Rating	Likes Count
count	13	11	13
mean	$ 123.46	6.7	19.8
min	$ 59.00	5.1	1
25%	$ 79.00	6.1	2
50%	$ 98.00	6.6	14
75%	$ 124.00	7.3	26
max	$ 348.00	8.2	59

Yellow Cluster

The final and smallest cluster was the yellow cluster. There were only 5 hotels in this cluster. While there were only a few hotels, there was a decent variety of restaurants with some places to shop (Figure 8). In addition, while there were at most two event spaces in this cluster, the prices for the hotels were less expensive compared to the red cluster which also had event spaces.

Figure 8. Total Number of Venue Types for Hotels in the Yellow Cluster



The hotels ranged in price from $80 to $119 with an average of $93.80 (Table 5). Half of the hotels in this cluster were $87 or less. The average rating for hotels in this cluster was 6.4, and the average number of likes was 23.

Table 5. Summary Statistics for Hotels in the Yellow Cluster on Price, Rating, and Likes Count

	Price	Rating	Likes Count
count	5	4	5
mean	$ 93.80	6.4	23.4
min	$ 80.00	5.4	3
25%	$ 87.00	5.9	5
50%	$ 87.00	6.4	23
75%	$ 96.00	6.9	34
max	$ 119.00	7.6	52

Discussion

The results of this analysis show that the combination of Foursquare data and hotel price data can provide additional information to travelers to help them make informed decisions on the best place to stay for their trips. Toronto, Canada was used as a test case in this analysis to determine if it was possible to segment hotels based on the types of venues near hotels and then provide travelers with additional information about estimated hotel price, ratings, and likes.

Hotels in Toronto fell into four different groups. A majority of the hotels (66%) were in the blue cluster which had a lot of variety in terms of eating, drinking, and things to do within walking distance. Travelers who may not want to rent a car and plan to stay within walking distance of their hotel should choose hotels in the blue cluster. There was a wide range of hotel prices and locations within the blue cluster giving travelers with different sized budgets access to these hotels.

Eighteen percent of hotels fell within the purple cluster. These hotels also had some variety for eating and caffinated drinks. These hotels may be ideal for short work trips where a traveler just needs some of the basics (e.g., food and drink) for their stay. These hotels ranged in the lower end of hotel prices.

Eleven percent of hotels fell in the red cluster. These hotels had plenty of coffee shops, but only limited types of places to eat. If someone had to stay in a hotel in this cluster, whether because hotels in the other cluster were sold out or they wanted to be near an event venue, it would help them to know ahead of time that they may need a rental car or some sort of transportation for any dining, drinking, or shopping needs. The price range for these hotels was almost as big as hotels in the blue cluster. The higher priced hotels may be due to their proximity to event venues making it even more important for travelers to know that while they are paying a high price for a hotel, there are few venues within walking distance and little variety.

The fourth and smallest group consisted of hotels (4%) in the yellow cluster. Surprisingly, these hotels had a variety in terms of eating and shopping and were in the lower range of hotel prices. Most of the hotels in this cluster were near the airport, which means there is some variety of places to eat within walking distance when a traveler wants to stay near the airport.

Conclusion

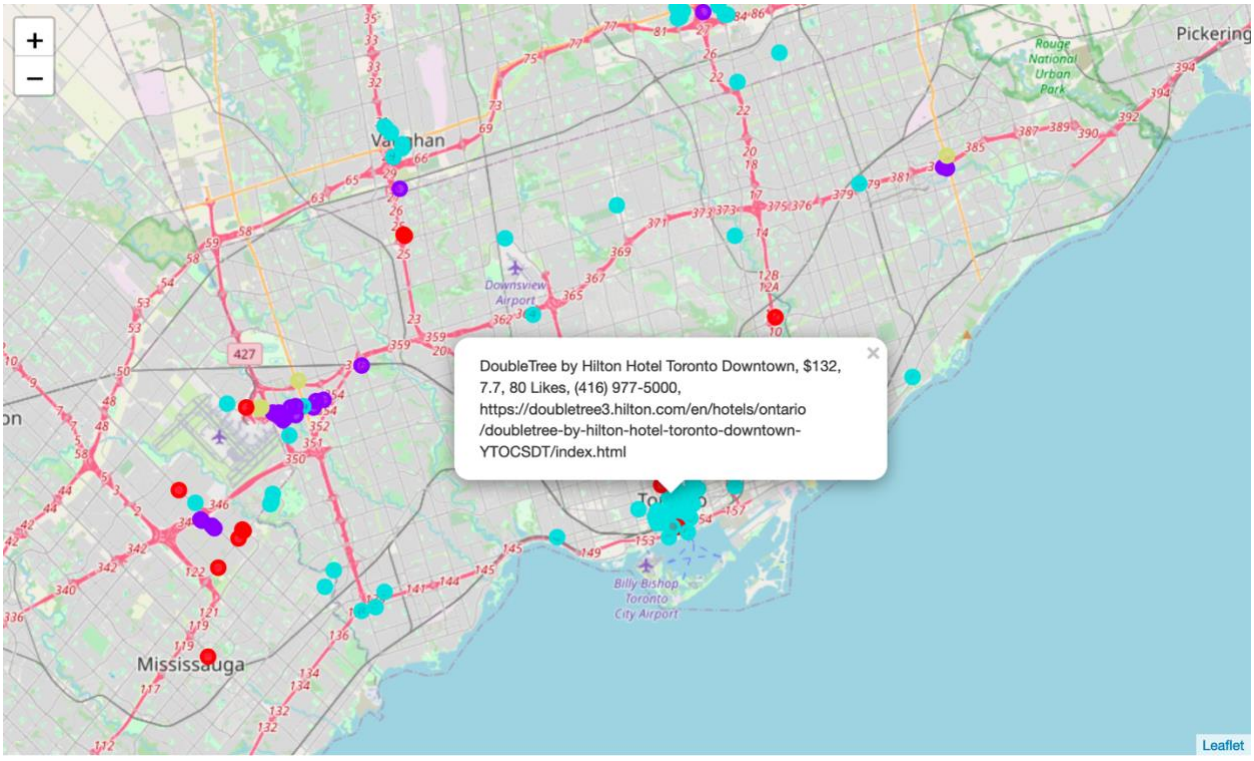
The purpose of this project was to test out a process that would help travelers choose the best hotel for their needs when planning a trip to another city. This project tested out the process with hotels in Toronto, Canada, using data from Hotels.com and Foursquare.

Based on the results from this project, a traveler coming to Toronto will probably want to focus on the hotels in the blue cluster as their first choice. Hotels in this cluster are located in various places throughout Toronto and have a variety of prices to appeal to any budget. On average, they are more highly rated and have more likes than hotels in the other clusters. These hotels also have a variety of drinking, dining, shopping, and activities within walking distance.

If a traveler isn't able to book a hotel within the blue cluster, the next best cluster of hotels would be the red ones. These had the second highest average ratings and a range of prices that would appeal to any budget. The one downside to hotels in this cluster was that there were not a lot of places to eat nearby or things to do. A traveler staying here would need to have some transportation if they wanted some variety in drinking, dining, shopping, and/or activities.

This would be a useful feature for travel companies to offer their clients. By using a combination of detailed Foursquare data and hotel price data, they could provide an interactive map that shows the different hotel clusters (Figure 9). The map provides an easy way for travelers to focus on a particular area and then choose a hotel where a pop-up reveals the name, price, rating, likes, hotel phone, and hotel URL.

Figure 9. Map of Toronto Hotels by Cluster with Interactive Pop-Up Example



In addition, if travel companies also offered a snapshot of the different venue categories associated with each cluster of hotels (Figure 10), travelers could easily make informed decisions about where to stay and whether or not they need additional transportation during their stay while planning for their trip. By combining existing resources (e.g. hotel websites and Foursquare data) and analyzing the data with machine learning algorithms, travelers can fully benefit during the planning process and may likely enjoy their stay more than if they didn't have these additional pieces of information. This will increase customers for both the hotels and nearby venues and potentially increase hotel ratings and likes.

Figure 10. Nearby Venue Categories by Hotel Cluster