1 Introduction

1.1 Problem Statement

Nowadays, travel and exploration are a part of everyday life. People may travel for a myriad of reasons like work, education, visiting friends and families, leisure, etc. They often find themselves exploring new areas of their towns, cities, countries or even the world. Though information is now available easily people spend a large amount of time rummaging through the data, without getting relevant results. This recommendation engine aims to make it easier for the wanderlusts by recommending venues that they could like in any area of the world based on what they have liked earlier on Four Square.

Also, local businesses like hotels, can provide more personalized service by understanding their customer preferences and making customer specific recommendations.

1.2 Target audience

This engine can be used by anyone who wishes to explore venues of interest for themselves or for someone that they are conducting this researchfilt for.

2 Data Acquisition

2.1 Data Sources

The primary source of data for this project is Foursquare. Foursquare Places API offers real-time access to Foursquare's global database of rich venue data and user content.

The following API Endpoints have been specifically used to extract the data used in this project:

Method	End Point Group	End Point	Usage
GET	venues	explore	Get Venue data for venues in an area
GET	users	lists	Get a User's lists
GET	lists	details	Get details of a list

2.2 Data Wrangling

The following actions were taken to prepare the data for analysis:

- Liked Venues Dataset
    - There is no straightforward way of extracting the venues that have been liked by a user. Since, this was absolutely essential to create a user profile of features that a user likes a workaround had to to be devised. Foursquare creates a list of places a user likes and add's it to the users list. Hence, this list was first extracted using the users->lists endpoint, and then the venue details were extracted using the lists->details endpoint.
    - Only relevant details like User First Name, User Last Name, Venue ID, Venue Name, Venue Latitude, Venue Longitude, Venue Category were parsed from the json response. All other data was dropped.
- Test Venues Dataset
    - The venues->explore was used to find the list of venues in a specific area by passing the latitude and longitude of the area of interest. Standard geocoder libraries were used to get the geographic location data. This data formed the Test Venues dataset.
    - Only relevant details like Neighborhood Name, Neighborhood Latitude, Neighborhood Longitude, Venue ID, Venue Name, Venue Latitude, Venue Longitude, Venue Category were parsed from the json response. All other data was dropped.
- The venue categories were used for creating the user's feature profile. Some of the venue categories were same as the field names (like Neighborhood). These categories were dropped from both the Liked Venues and Test Venues.

2.3 Feature Selection:

The primary feature used for the analysis was the Venue Categories. This feature was selected as this feature had the most complete data. Also, Venue Category was readily available for all venues in the free version of the data set. Other features or attributes that could be used were only available in premium end points.

3 Methodology:

3.1 Solution Strategy:

The recommendation engine uses a content-based recommendation strategy

3.2 Solution Methodology:

The content-based recommendation was implemented using the algorithm explained below:

Neither of the unique set of categories for the above data sets was a superset of categories that could be used for one-hot encoding. Creating a superset of all unique categories in the Foursquare Dataset is not available and not feasible, because of the huge number of categories involved. Hence, the union of the unique categories was decided to be used for one-hot encoding. The following steps were followed to get to this result.

- Create a combined dataset of all venues by combining the data from:
  - Liked Venues by User (liked_venues)
  - Target Venues to score/rank (test_venues)
- Use the combined categories data for one hot encoding by Venue Category.
- Add the Venue Id and User Id fields so that the data can be appropriately merged with the liked venues and test venues
- Create a User Profile for the user
  - Join the liked venues with the one-hot encoding data by Venue Category
  - Since, the user either likes or dislikes the venue there is no need to multiply by a weight.
  - Sum up the values for all rows grouping by UserId. (Mean could be used here too)
  - Transpose and retain the series

	Value
American Restaurant	1
Bakery	2
Bank	0
Café	0
Cantonese Restaurant	0
Chiropractor	0
Coffee Shop	1
Cosmetics Shop	2
Department Store	1
Design Studio	1

Diner	1
Elementary School	1
Fried Chicken Joint	1
Gastropub	0
Greek Restaurant	1
Grocery Store	2
Gym	0
Halal Restaurant	1
Home (private)	2
Japanese Restaurant	0
Laundry Service	1
Mexican Restaurant	1
Nail Salon	1
Pharmacy	0

Pizza Place	2
Restaurant	0
Salon / Barbershop	1
Sandwich Place	1
School	1
Seafood Restaurant	1
Shopping Mall	1
Southern / Soul Food Restaurant	1
Supermarket	1
Sushi Restaurant	0
Tea Room	0
Thai Restaurant	0
Theme Park Ride / Attraction	1
Wings Joint	1
Yoga Studio	0

*** These are partial screenshots and do not display the full list*

- Create a matrix of features for the target venues to rank by
    - Join the test venues with the one-hot encoding data by Venue Category
    - Drop the unnecessary columns

VenueId	American Restaurant	Bakery	Bank	Café	Cantonese Restaurant	Chiropractor	Coffee Shop	Cosmetics Shop	Department Store	Design Studio	Dir
52b1e6a8498e43e8f94b4b25	0	0	0	1	0	0	0	0	0	0	
526ecc2b498edb771548d405	0	0	0	0	0	0	0	0	0	0	
4bee9a19e8c3c92897d89892	0	0	0	0	0	1	0	0	0	0	
4af379edf964a52002ee21e3	0	0	0	0	0	0	0	0	0	0	
4b0dc067f964a5206b4f23e3	0	0	0	0	0	0	0	0	0	0	

*** This is a partial screenshot and does not display the full feature table*

- Calculate the Venue rank/score by multiplying the User Profile with the Feature Matrix and taking a weighted average.

```
VenueId
5be9d391d41bb700391ae899    0.064516
4b50d181f964a520853327e3    0.064516
4aea3276f964a52019ba21e3    0.064516
4c1863af4ff90f47f2670e49    0.064516
4c02943e0d0e0f475549019a    0.032258
4adf0a84f964a5203a7721e3    0.032258
4ae71ecef964a5209ea821e3    0.032258
4af5d177f964a52042fd21e3    0.032258
588d1d78a149261a47e1236d    0.032258
4b4b8601f964a5201c9f26e3    0.032258
4ade2eeff964a520b07321e3    0.032258
Name: Score, dtype: float64
```

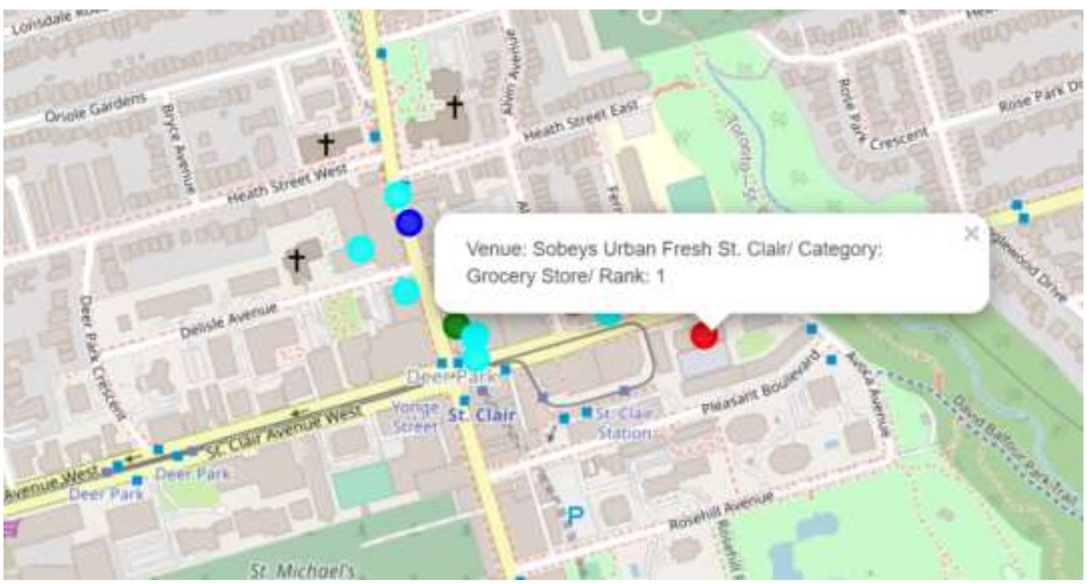
*** Representation of how the Venues are scored*

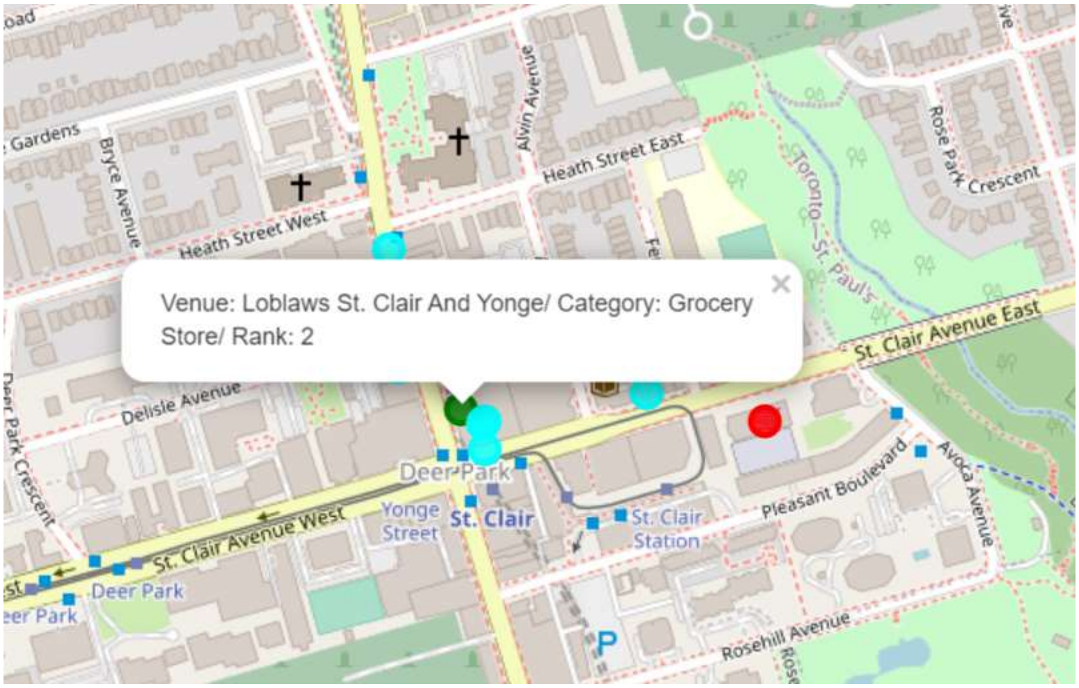
- Join the ranks/scores back to the test_venues using the Venue Id and display the top-n recommendations. For the purposes of this exercise the number of recommendations was limited to 10.

	Neighborhood	VenueId	Venue	Venue Latitude	Venue Longitude	Venue Category	Score
0	Eaton Center, Toronto, ON	4aea3276f964a52019ba21e3	Sobeys Urban Fresh St. Clair	43.688479	-79.390769	Grocery Store	0.064516
1	Eaton Center, Toronto, ON	5ba9d391d41bb700391ae899	Loblaws St. Clair And Yonge	43.688569	-79.394034	Grocery Store	0.064516
2	Eaton Center, Toronto, ON	4c1863af4ff90f4712670e49	Pizza Pizza	43.689555	-79.394657	Pizza Place	0.064516
3	Eaton Center, Toronto, ON	4b50d181f964a520853327e3	Loblaws	43.688484	-79.393781	Grocery Store	0.064516
4	Eaton Center, Toronto, ON	4ade2eeff964a520b07321e3	Starbucks	43.688467	-79.393801	Coffee Shop	0.032258

*** This is a partial screenshot and does not display the full recommendation list*

- Visualize the results on a map. To make it easier to spot the top 3 results, they are added to the map with their own marker color.
    - Marker for Rank 1 is displayed in red
    - Marker for Rank 2 is displayed in green
    - Marker for Rank 3 is displayed in blue
    - All other ranks are displayed in cyan




Venue: Loblaws St. Clair And Yonge/ Category: Grocery Store/ Rank: 2
Heath Street West
Heath Street East
Alvin Avenue
Bryce Avenue
Rose Park Crescent
Toronto–St. Paul's
St. Clair Avenue East
Delisle Avenue
Deer Park Crescent
Deer Park
Yonge Street
St. Clair
St. Clair Station
Pleasant Boulevard
Avoca Avenue
St. Clair Avenue West
Rosehill Avenue


Venue: Pizza Pizza/ Category: Pizza Place/ Rank: 3
Oriole Gardens
Alvin Avenue
Toronto–St. Paul's
St. Clair Avenue West
Delisle Avenue
Deer Park Crescent
Deer Park
Yonge Street
St. Clair
St. Clair Station
Pleasant Boulevard
Rosehill Avenue

4 Conclusion

The venue recommendation engine is effective at filtering locations that match the profile of venue categories that the user has liked. As described in the problem statement this can be very useful to the target audience of this solution

5 Limitations and Future Enhancements

At the same time, I would like to note that this solution is only a basic framework and has certain limitations:

- Only Venue categories were used as a feature, which could be a limiting factor in creating a complete user profile. A more complete profile would benefit from use of attributes that can be extracted from premium endpoints.
- The content-based recommendation approach is heavily dependent on historical likes and does not address new recommendations that a user could try. A collaborative filtering strategy can help address this issue.
- The model can be fine-tuned to focus on specific categories and areas of interest e.g. food, tourist attractions, etc.
-