# Introduction

## 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.

## 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.

# Data Acquisition

## 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 |
|  |  |  |  |

## 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.

## 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.

# Methodology:

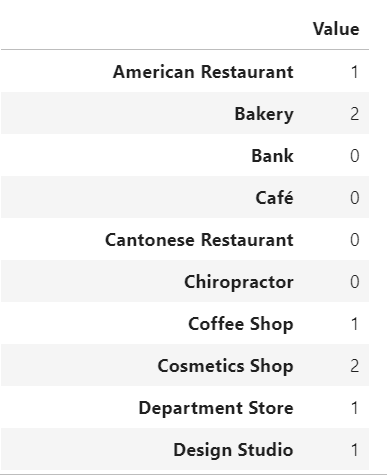
## Solution Strategy:

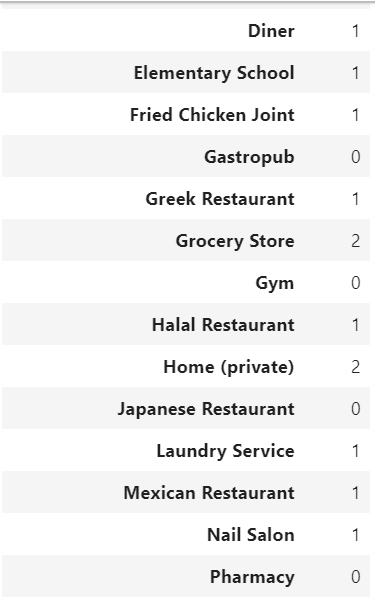
The recommendation engine uses a content-based recommendation strategy

## 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

.  

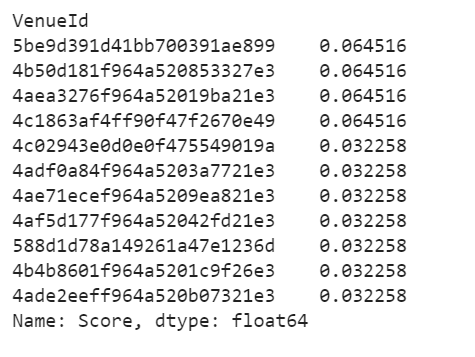
*\*\* These are partial screenshots and do not display the full list*

* 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



*\*\* 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.



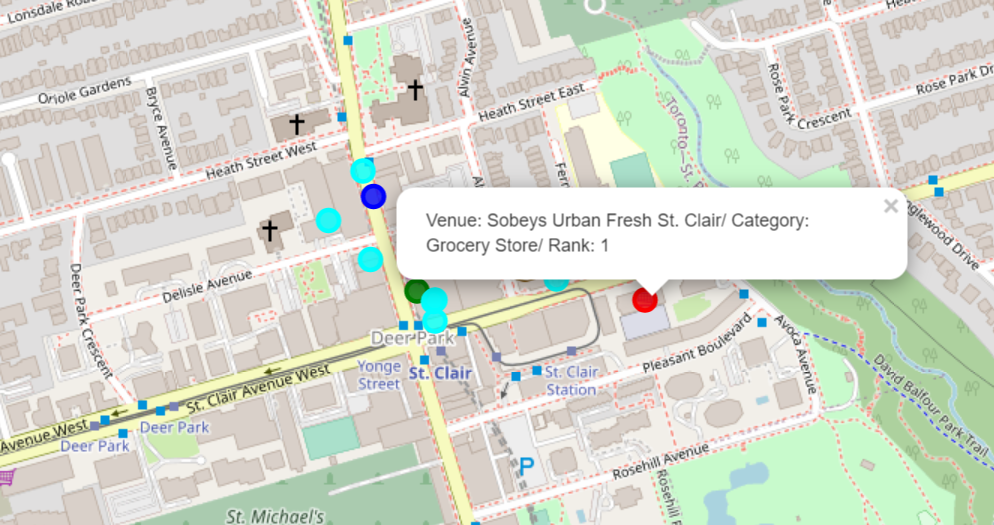
*\*\* Representation of how the Venues are scored*

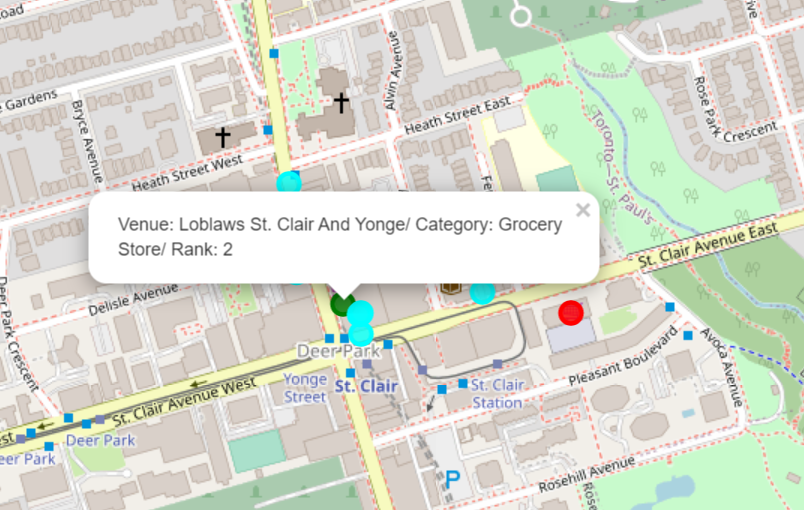
* Join the ranks/scores back to the test\_venues using the Venue Id and display the top-n recommendations. For the purposes of this exercise the number of recommendations was limited to 10.

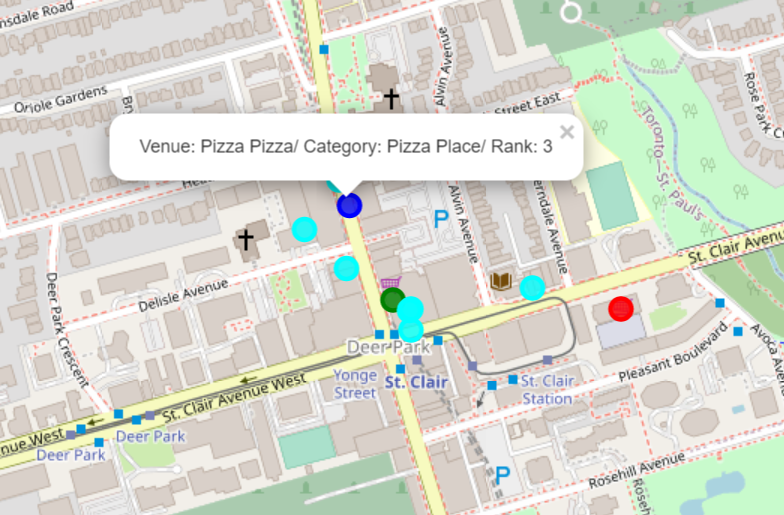


*\*\* This is a partial screenshot and does not display the full recommendation list*

* 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







# 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

# 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.