Coursera Capstone Project - Toronto Neighborhood Recommender System

Alejandro Rojas Benítez
Medellín, Colombia
alejorojasb98@gmail.com

***Abstract*— When people move to a new city and they cannot find any suggestions on what neighborhoods they should move in or start a new business, that is when they can use this application, where they just type what type of venues they would like along with a score between 1 and 10 and they will receive a recommendation. We use a basic recommender system with some processed data from Foursquare API and information from Toronto, Canada. The algorithm works and we tested it on 3 different user inputs: 2 different tourists and a student.**

I. INTRODUCTION

Imagine we want to create an app that is able to recomend neighborhoods based on specific requests: Population, employers, type of venues? For example, someone who wants to start a new business, maybe a new italian restaurant?: High population, low amount of restaurants so less competition and a decent amount of parks to get customers when they go out for a walk.

Currently, there are available different apps to check specific recomendatios about places at the neighborhoods of a city (nextdoor, google maps, even foursquare). But for users who are moving into new cities, where they have zero to none clue about the new places they will be living at for the next few months or even years, we want to bring a solution to them, which will help in the process of selecting a neighborhood or at least to get a list of a couple neighborhoods which follow some of their requirements. It doesn't matter if they want it to start a business, live, invest, or whatever idea they may have.

II. DATA

On this initial approach we will work using Toronto neighborhoods data we found on kaggle: Kaggle data which uses data from the open source data portal of Toronto. Here we have different information for each neighborhood such as:

- Population
- Number of educated people
- Population with ages between 15 and 45
- Number of employers
- Coordinates: which will enable us to do some queries on the foursquare API.

Now, adding information about the venues from foursquare we can manually group the 200+ categories into the following 18 ones:

- Arts
- Bank
- Bar
- Beauty
- Cafe
- Community Center
- Education
- Fast Food
- Gas Station
- Groceries
- Hotel
- Park
- Pharmacy
- Public Transport
- Restaurant
- Road
- Sports
- Store

Finally, we will append some crime information so users could even see that at the final result of the app. We imagine that almost any person would prefer a place with zero crime so this columns could help to select one final place after receiving multiple results on the app. This information will be obtained from the Toronto crime data.

III. DATA ANALYSIS

For this section we will plot histograms of the frequency of all venues in our custom neighborhoods and try a kmeans clustering algorithm, if a lot of them are grouped in the same group we could end up having a problem of them being to similar.

First, we can see 1 that the most common venues are Stores, Restaurants, Fast Food, Cafes, even Parks. This may lead us to think that when users look for specific venues they algorithm may stick with the same couple neighborhoods. Still we believe on the simple recommender system algorithm for our initial version of the app.

On the other hand, we tested a clustering algorithm to check how a simple k means would group our different neighborhoods. We would like to have different balanced groups and not almost all of the neighborhoods assigned to the same cluster (to show at least some variability between our rows). We can see the clustering results on figure

Some of the remarkable information we found about the clusters is:

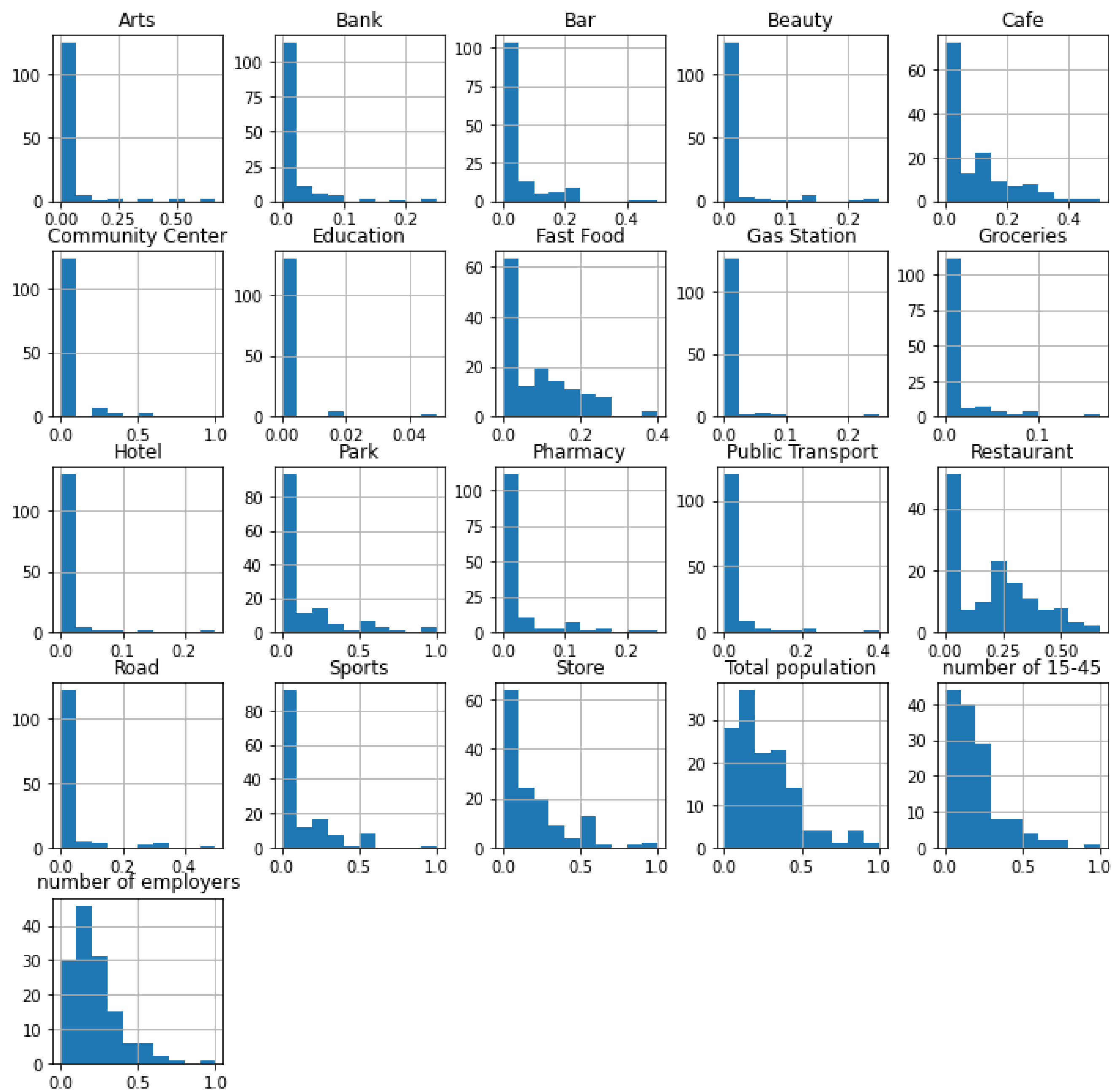

Fig. 1: Frequencies of venues on each neighborhood.

- Clusters 0, 1 and 3 have high values on Arts compared to Cluster 2.
- Cluster 3 has no banks
- Cluster 1 has higher fast food presence
- Even though we didn't cluster using the population variables, Cluster 2 has the highest population average between all 4 clusters.

This clustering is valuable, but we will not include it on the initial version of this recommender system.

IV. METHODOLOGY

This recommender system will start as an minimum viable product, using the algorithm showed by the IBM course "Machine Learning with Python" on the content based recommender systems.

First, we will have an initial matrix $M_{(nxv)}$ where every row represents a neighborhood and each column contains the proportion of the venue category for that neighborhood, n is the number of neighborhoods and v the number of venue categories. For example we can have "Toronto neighborhood one" where it has 0.6 for restaurants, 0.2 for parks and 0 on the rest of the venue categories, of course this is only an example.

Now, our user will create an input with the desired venue values which is a vector I_{1xk} where $k \leq v$, the venue

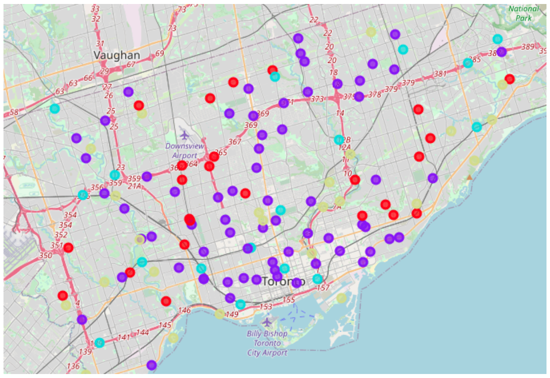

Fig. 2: Clustering results.

categories doesn't need to be on the same order as the on the matrix M. Now we will organize the input for it to match the order of the venue categories from our data matrix, and create a new vector $\hat{I}_{1xv}$ where every $I_i \leq 10 \quad \forall i \leq v$. 10 will indicate that the user has great interest on that category and 0 that there is a low interest.

Then, our final recommendation R will be calculated as

$$R = \sum_{i=1}^{n} \frac{(M \circ \hat{I})_i}{|\hat{I}|}$$

where $\circ$ represents the element wise multiplication, we sum all the column values and divide by the total ratings of our user, leaving us with a $nx1$ vector with a value for each neighborhood. Then we will show the top 3 results in our app, and some of the extra information we gathered up on the Data section (crime rates, population).

V. RESULTS

We will test 3 different inputs, labeling them as **Arts tourist**, **Student** and **Party tourist**. They will have the following inputs:

- Arts tourist: Arts=10, Bank=1, Restaurant=6, Sports=8
- Student: Public Transport=10, Education=10, Community Center=6, Groceries=6
- Party tourist: Bar=10, Store=10, Restaurant=10

We included the top 3 neighborhoods for each of the user inputs, they can be seen on image 3. It can be seen that the 3 different users had completely different recommendations and our app is able to include different information such as population and crime rates. This should give them extra information on what place could fit the best their needs while staying or moving in to the city.

VI. DISCUSSION AND CONCLUSIONS

We built an initial version of a neighborhood recommender system for the city of Toronto, using a basic recommender system algorithm along with some data cleaning from foursquare API. We tested it on 3 different inputs: a tourist (heavy desire on arts, sports, banks), a student (heavy desire on education, community centers, public transport and

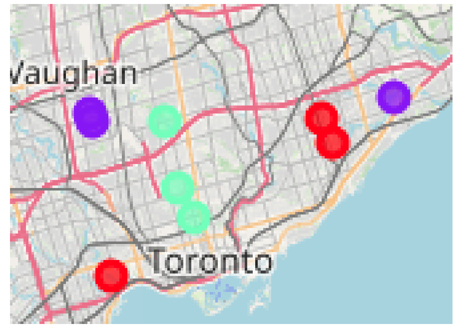

Fig. 3: Recommended neighborhoods for the different users: Arts tourist = purple, Student = cyan and Party tourist = red.

groceries) and another type of tourist (bar, stores and restaurants) and the overall results showed different neighborhoods that could fit them.

The algorithm works but it has a lot of room to generate more accurate results, along with including more information of what a user may desire (house prices, real state liquidity, car insurance prices, and more).

As future work I propose including a friendly user interface where users can select the wanted categories, more cities and a web app.

REFERENCES

[1] P. Juszczuk and L. Kruś, "A fuzzy multicriteria approach for the trading systems on the Forex Market," *Multiple Criteria Decision Making*, vol. 14, pp. 29–43, 2019.
[2] IBM, "Machine Learning with Python - Coursera."