Neighborhood Recommender System: Comparing Neighborhoods Between Cities

Hasmik Grigoryan

August 2019

This project was completed for IBM Data Science Professional Certificate specialization on Coursera.[2] This serves as my final project for the Capstone. I will explore neighborhoods in Chicago and New York City and create a recommender system for neighborhoods between these cities.

1 Introduction / Business Insight

Most big enough cities have similar sets of neighborhoods; the homey ones, the ones great for night life and entertainment, the ones with the most museums, etc. In my experience, it takes some time before you figure out what your favorite ones are when moving into a new city. For this reason, I will try to make a recommender system for neighborhoods. I will be exploring content-based recommendation and clustering methods for this purpose.

While this tool may be particularly useful to individuals who are looking to relocate between Chicago and New York, it can be of interest to realtors and property management companies. For instance, a realtor can use this tool to provide better recommendations for their clients moving between NYC and Chicago. Property management companies may use the content for better targeted advertising and for analyzing their clientele's preferences.

2 Data

For this project I will be mainly using Foursquare API [1], the list of neighborhoods scraped from Wikipedia [3] and Geopy for address search. Foursquare provides an easy way for exploring venues in proximity of given locations. These venues come with a lot of accompanying data, including categories, ratings, and names. I will be using this data for feature engineering per neighborhood. For instance, I can design categorical features that indicate most common category of venue within the neighborhood. Similarly, we can look at the categories with highest cumulative ratings. The Wikipedia data will be scraped for neighborhood list using Pandas.

My attempt is to make this tool as general as possible, so that it can work for any city that Foursquare data is available for. However, a list of neighborhoods of a city will be necessary. For this reason, I will be making the tool for Chicago and New York only. But the tool will work for any city when the data on neighborhoods in appropriate format is provided.

3 Methodology

The project consists of 5 main parts.

- Scraping neighborhood names and locations
- Cleaning neighborhood data
- Scraping venue data per neighborhood
- Feature engineering from the venue data
- Training ML models and presenting results

3.1 Scraping neighborhood data

Python Pandas library was used to scrape initial list of names of the neighborhoods in Chicago and New York. Once I got the list of neighborhoods, I used geopy in order to find the latitude and longitude values of all of the neighborhoods. These were put in a pandas dataframe.

3.2 Cleaning neighborhood data

Geopy returned NULL values for some of the neighborhoods. Further investigation showed that this either happened when neighborhoods weren't named right (e.g. used nicknames instead of official names) or weren't in fact neighborhoods. Sometimes a collection of residential commons or a small campus made it to the list. These were dropped after being examined.
At this point I was able to visualize a map of the neighborhoods for each city. I noticed that the location that geopy returned for some of the neighborhoods (around 5 for each city) were not in fact in the city. I examined these further. I googled these neighborhoods, the ones that resulted in valid google maps searches were added to the dataframe together with the location manually extracted from google maps. The rest were dropped. A final visualization and visual exploration of the map showed that the neighborhoods extracted were correct.

3.3 Scraping venue data

I used geopy to look up for nearby venues at each neighborhood. In addition to the location the query takes in 2 values: limit to the number of venues to return and radius for proximity of venues. For the radius I ended up choosing 2 kilometers. 2 kilometers is supposed to take around 20 minutes to walk. This means that when the tool recommends a neighborhood it ensures that the venues the recommendation is based on are within walking distance.

3.4 Feature engineering

For feature engineering I extracted the categories of each venue for the neighborhoods. So the feature ended up being categories with the value indicating number of venues of said category within a neighborhood. I ended up with 484 different categories. I read through all of them and noticed that quite a few could in fact belong to the same category for instance cafe and coffee shop. After joining a few such categories I ended up 469 features.

3.5 The ML models

K-Means

The initial model that I used for the analysis was K-Means clustering. I chose the numbers of clusters experimentally and settled on 20. I then visualized the neighborhoods colored by clusters. This is a good tool for exploring the neighborhoods in general. However, what if I know my favorite neighborhood in one of the cities and I'm curious to find what neighborhood in the other is most similar to it?

Content Based recommender

For particular recommendations I used a recommender based on cosine similarity. Given a neighborhood I then use the above extracted features to find most similar neighborhoods in the same or the other city. These were then visualized based on a Green-to-red color scale where neighborhoods in green are most similar to the one of interest.

4 Results

The clustering shows that most clustered neighborhoods tend to be geographically nearby within the city. Because the clustering was done on all of the neighborhoods both in Chicago and NYC, one can look for neighborhoods in both cities that belong to the same cluster for recommendation. However, this tool does not provide a more concrete recommendation for someone who's looking for a quick list instead of exploration. In this case, the cosine-similarity works best. For a sanity check, I tried seeing what recommendations I would get based on my own preferences. Neighborhoods downtown in Chicago tended to result in recommended neighborhoods downtown in New York. Similarly, artsy neighborhoods such as SoHo in NYC recommended artsy ones such as Wicker Park in Chicago.

5 Discussion

I believe joining some of the features in order to get less features might be a good idea. Some of the features can get very specific and further analysis would need to be done in order to decide whether that is beneficial. For instance, should restaurants be abbreviated to just food, or should cuisine matter? If the latter, how specific should cuisines be?

Because the purpose of this tool is recommending neighborhoods further analysis could be done to provide explanations for the recommendations. Right now, each recommendation comes with a similarity score. However, given the simple design of the features, one could provide a visualization of similarity. Example venues from the most influential features could also be a good idea.

6 Conclusion

Data was gathered on common categories of venues within each neighborhood in Chicago and New York City. Clustering of these neighborhoods shows that similar neighborhoods do end up being geographically nearby within each city. The cosine similarity suggests a useful tool for suggesting starting points for exploring new neighborhoods in one of the cities based on a favorite neighborhood in the other city. This tool is better used interactively so I highly recommend looking into the code.

References

[1] Foursquare. https://foursquare.com/.

[2] Ibm data science professional certificate. https://www.coursera.org/professionalcertificates/ibmdatascience.

[3] Lists of neighborhoods by city. https://en.wikipedia.org/wiki/Lists_of_neighborhoods_by_city.