No Place Like Home.

Recommendation system for relocations.

Andrey Krupnov

March 25, 2020

Introduction/Business Problem

According to the latest information published by the U.S. Census Bureau the percentage of people that move every year equates to 14% of the population (or roughly 40 million). According to U.N. data the number of international migrants worldwide has continued to grow over the past seventeen years, reaching 258 million in 2017, up from 248 million in 2015, 220 million in 2010, 191 million in 2005 and 173 million in 2000. These people relocating in search of new jobs, career opportunities, better living conditions etc. They leave neighbourhoods they were used to and come to mostly unknown places.

How can we make relocations more comfortable to reduce inevitable stress of coming to a new city andmake people fill people as much at home as possible? This will surely be useful for their productivily, health and wellbeing.

Imagine you want to move to another city to join a team of Data Scientists working on a project of your dreams (imagine there's no cononavirus quarantines either). You like the neighbourhood you live in and you want to have similar set of venues (shops, caffees etc.) around your new home. So, where it should be? In what neighbourhood should you look for an apartment to rent when you come to your new city?

Well, the answer can be obtained using the recommendation system 'There's No Place Like Home'.

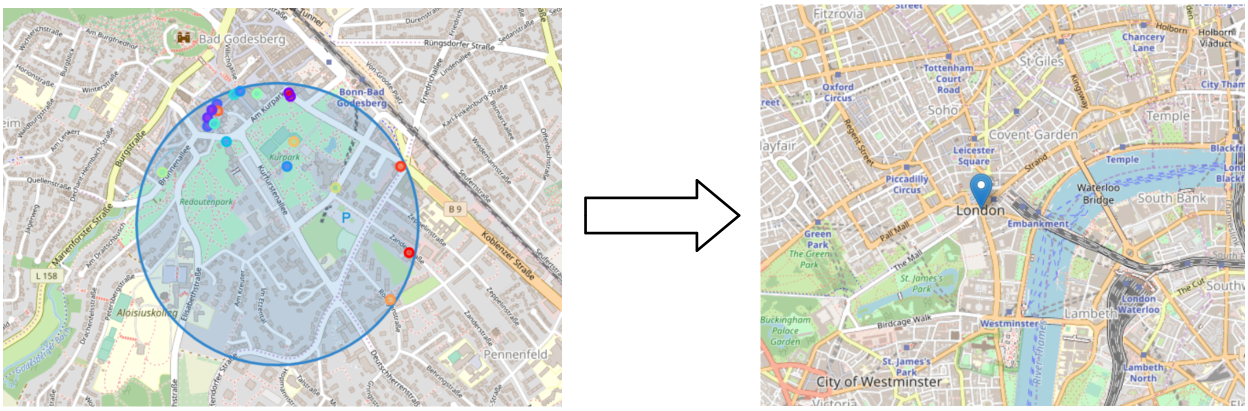
User Experience and Data Sources.

First, a user should submit his/her current address (or, if user is not quite happy with his/her current neighbourhood, he/she may submit any other known address that seems attractive enough). System is not limited to any geographical location, the only limitation is the amount of information in Foursquare base. Geographical coordinates of user's home will be acquired using Nominatim

Geocoder and then Foursquare query will get us a list of venues situated around it.

Then, a user should provide the name of the city he/she is to move to. City map will be divided to squares and then the system will determine the one that's most similar to user's present address according to the set of venues in it. Then Foursquare queries will be used to acquire data on venues surrouding given location and build a dataset to perform collaborative filtering to figure out the perfect location for our user to move to.

In the example I use an address in Bad Godesberg as home address (I used to live there for 4 years in my childhood and I like the place still) and London as a target city (kind of compensation because my most recent business trip cancelled because of COVID-19 was supposed to be to London). But every user is free to enter data of any two places of his/her choice.



The result would be presented to user in the form of an address of the center of the most similar square (reverse geocoder will provide an address of the location (based on it's coordinates) that was choosen as the most suitable for relocation), so that user would have a starting point to look for a property to rent. He/she will be also able to see vusualization of recommended neighbourhood with all the venues in it.

As no one can guarantee that user can find (meaning to rent or buy) a new home or apartment at the exact address at the center of the neighbourhood the system has found, it shouldn't be mathematically precise. We need to find just the right neighbourhood.

Data preparation

For the analysis the system will need the following data from the home neighbourhood:

#	Field	Description
1	Venue	Names of every venue in a neighbourhood. Used for informational purposed only.
2	Venue Latitude	Used to determine location on the map when visualizing results

3	Venue Longitude	Used to determine location on the map when visualizing results
4	Venue Category	Used to filter the results and build weights vector
5	CategoryID	Used to form queries in the new city to filter out all venues categories that don't present in home neighbourhood

The final home heighbourhood portrait will look like this:

	Neighborhood	Burger Joint	Clothing Store	Drugstore	Event Space	Fish Market	Flower Shop	French Restaurant	Hotel	Italian Restaurant	Metro Station	Park	Pizza Place	Shopping Mall	Theater	Tunnel	Video Store
0	Home	0.1	0.1	0.1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.1	0.05	0.05

Because in Foursquare queries if specifying a top-level category, all sub-categories will also match the query, we need to additionally filter query results to get rid of venues categories non-existent in original.

Target city is divided to equally sized areas partially overlapping one another. Total number of adeas should cover most part of big city square (in the example there are 15x15 = 225 circles covering approx. 100 sq. km). Partial overlapping is used to solve 'one more step' problem, when some required venues are located just beyond the border of Area 1 and some - just in front of it. In a case like this, Area 2 that overlaps a part of Area 1 from that particular side, will contain all of them.

For target region I make queries limiting results to the list of venue categoty IDs present at the departure point. Thus I achieve two goals. First, I select only the information relevant to the task of comparing new neighbourhoods with the old one.

Second, I deal with a problem of limit for number of results of one SEARCH query. Foursquare applies a limit so you can't get more than 50 venues within 1 query and daily number of queries is limited as well. For a big city like London you should make approx. 120-150 queries to cover the substantial part of the map.

For target regions I retrieve the following data:

#	Field	Description
1	Venue	Names of every venue in a neighbourhood. Used for informational purposed only.
2	Venue Latitude	Used to determine location on the map when visualizing results
3	Venue Longitude	Used to determine location on the map when visualizing results
4	Venue Category	Used to filter the results and build decision matrix

So actually it's all about venues categories in a target city.

I will also need number of venues to set the target for recommendation system. If the home neighbourhood looks like this:

	Venue Category	Venue	Venue Latitude	Venue Longitude	CategoryID
0	Burger Joint	2	2	2	2
1	Clothing Store	2	2	2	2
2	Drugstore	2	2	2	2
3	Theater	2	2	2	2
4	Event Space	1	1	1	1

then the system should prefer locations with 2 burger joints to places with 1.

I use One hot encoding to build a matrix of venues at every circle within target city:

	Neighborhood	Burger Joint	Clothing Store	Event Space	Fish Market	Flower Shop	French Restaurant	Hotel	Italian Restaurant	Metro Station	Park	Pizza Place	Shopping Mall	Theater	Tunnel	Video Store
0	Region #0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	Region #0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
4	Region #1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
5	Region #1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
6	Region #1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

and then I group venues belonging to different circles and count their numbers

	Burger Joint	Clothing Store	Event Space	Fish Market	Flower Shop	French Restaurant	Hotel	Italian Restaurant	Metro Station	Park	Pizza Place	Shopping Mall	Theater	Tunnel	Video Store
0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
1	0	0	1	0	0	0	2	0	0	1	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
3	2	0	4	0	1	0	5	9	1	5	1	0	3	0	0
4	2	2	1	0	0	1	9	7	2	4	1	2	1	0	0

There exists a problem: we should avoid situation when a heighbourhood with 0 venues in most categories and 50 venues in one of the categories will get high score. Similarity of neighbourhoods mean having at least 1 venue in each category rather than having tons of venues in one category.

To ensure this I modify data in target city using logarythm formula:

Data_new = ln (Data_old + 1)

So zeros will remain zeros, but system will penalize for not having a single venue in a category more than award for having extra.

	Neighborhood	Burger Joint	Clothing Store	Event Space	Fish Market	Flower Shop	French Restaurant	Hotel	Italian Restaurant	Metro Station	Park	Pizza Place	Shopping Mall	Theater	Tunnel
0	Region #0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.693147	0.000000	0.693147	0.000000	0.000000	0.000000	0.0
1	Region #1	0.000000	0.000000	0.693147	0.0	0.000000	0.000000	1.098612	0.000000	0.000000	0.693147	0.000000	0.000000	0.000000	0.0
2	Region #10	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	1.098612	0.000000	0.000000	0.000000	0.0
3	Region #100	1.098612	0.000000	1.609438	0.0	0.693147	0.000000	1.791759	2.302585	0.693147	1.791759	0.693147	0.000000	1.386294	0.0
4	Region #101	1.098612	1.098612	0.693147	0.0	0.000000	0.693147	2.302585	2.079442	1.098612	1.609438	0.693147	1.098612	0.693147	0.0

Recommendation system

To find a perfect place for relocation I use content-based recommendation system algorythm with home heighbourhood data as weights vector and target city matrix as values to be compared.

Similarity score for every circle is calculated:

```
Neighborhood
Region #174     1.124896
Region #113     1.101304
Region #128     1.009600
Region #130     1.009000
Region #99      0.996238
```

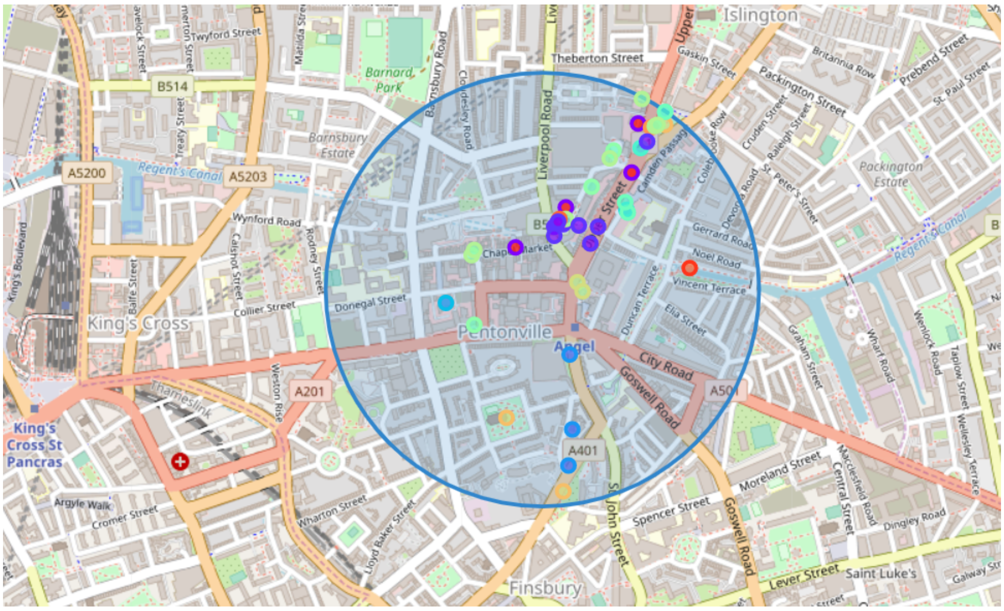
It's possible that there would be more than 1 circle with the same winning score (I experienced this situation when tuning the model). In this case the system will recommend any number of winning places.

An address of winning region(s) is being determined using reverse geolocator query:

```
Neighborhood Latitude      51.532730
Neighborhood Longitude     -0.107168
Name: Region #174, dtype: float64

Kaplan, 10-14, White Lion Street, Angel, Pentonville, London Borough of Islington, London, Greater London, England, N1 9PF, Uni
ted Kingdom
```

In the end we show winning region on the map (all venues are visualized with every venue category in different color):



Lastly we compare home and destination heighbourhoods side to side to show both their similarities and difference:

Venue Category	Venue	New Venue
Burger Joint	Rosa's Burger, Beef Buddies	Five Guys, Chapel Market Burger Stall (The Nak...
Clothing Store	H&M, C&A	Monsoon, Next, French Connection, H&M, Joy, GA...
Drugstore	dm-drogerie markt, Rossmann	NaN
Event Space	La Redoute	Lift
Fish Market	Fischgeschäft Stuch	Moxon's Fish Monger
Flower Shop	Blumen aus Holland	Angel Flowers
French Restaurant	Ruby's Baguetterie	Le Coq Epicer, Frederick's Bar & Restaurant, C...
Hotel	Insel Hotel Bonn	DoubleTree by Hilton, Premier Inn London Angel...
Italian Restaurant	Terra Vino	Don Matteos, Dolcetto, Bella Italia, La Forche...
Metro Station	U Stadthalle	Angel London Underground Station, Thoughts Of ...
Park	Kurpark	Myddelton Square, Spa Green, Islington Green
Pizza Place	Zum Steinofen	Franco Manca
Shopping Mall	Fronhofer Galeria	Angel Central, Argos
Theater	Kleines Theater Bad Godesberg, Kammerspiele	Sadler's Wells, Old Red Lion Pub and Theatre, ...
Tunnel	Bad Godesberger Tunnel	Islington Tunnel
Video Store	Empire Videothek	NaN

Conclusion and future improvements

’No Place Like Home’ recommendation system in it’s present form can predict target neighbourhood based only on surrounding venues, their categories and number. It can be useful for people willing to relocate (even from the same city) in saving them time and efforts choosing a good place to live. It can also be used just for fun as a test ’What neighbourhood in foreign city is muck like your current one?’

It may be improved with adding new factors such as real estate price level, ecological situation, traffic situation and so on.