Finding Sister/Twin cities based on preferred user criteria

1. Introduction

1.1 Background

Traveling can bring many opportunities to explore new culture and experience unknown world. However, if traveling for business or other reasons than exploring, sometimes it's nice to stay in a new place which could offer the same or similar attraction and facilities as traveler is used to in his/her home city. This lets person focus on business without too many distractions or discomfort.

1.2 Problem

With Britain living the EU, there is increased need for individuals in the EU to learn one of major European language other than English. The best way to achieve it, is to move to foreign country and get exposed to other language as much as possible in your daily life.

I decided to use machine learning to find "sister/twin" city to my home city in order to ensure that I won't be missing out some recreation facilities/landmarks. The main goal is to learn the language without changing the city environment and available facilities too much.

Criteria for "sister' city: Countries to consider are Germany, France, Italy, Spain. The city of choice must be similar to my home city in Outdoors & Recreation facilities/landmarks offered. Also, the city has to be big enough (population over 400000). The plan is to live in central area of the city and only facilities/landmarks located near the city center need to be considered.

1.3 Interest

The idea of finding sister/twin cities based on preferred facilities or infrastructure can be used by individuals or businesses looking to relocate or expand they businesses. If current location provides desirable conditions for living or successful business environment, it can be assumed that sister/twin city is likely to provide same or similar environment. It does not guarantee 100% success, but such tool can be used to mitigate the risks involved when choosing new places for expanation or relocation.

2. Data acquisition and cleaning

2.1 Data source

- Data on biggest cities in Europe with population will be taken from: http://worldpopulationreview.com/continents/cities-in-europe/
- Foursquare data based on every city location will be used to get all Outdoors & Recreation facilities/landmarks.
- Geocoding web service (Nominatim) will be used to locate the coordinates of the cities

- https://www.wikipedia.org/ will be used for cross referencing data acquired from above sources for quality assurance purposes.

2.2 Data cleaning

European City data was scraped from the web. Only cities in interest countries (Germany, France, Italy, Spain) with preferred size of population (over 400000 people) were selected for analysis.

Geographical location of cities was found using geocoding web service. The data was plotted on the map. It was noticed that Zaragoza city location wasn't correct. It was manually fixed based on information from Wikipedia.

2.3 Feature selection

Attractions/Facilities available in Vilnius were analyzed and feature set for cities comparison was selected based on users preferences. Feature set selected as follows:

Park, Plaza, Other Great Outdoors, Trail, Scenic Lookout, Beach, Historic Site, Pool, River, Athletics & Sports, Basketball Court, Playground, Fountain.

All Attractions/Facilities from Foursquare data base were filtered based on feature set selected.

3. Exploratory data analysis

Taking into consideration that the cities are different in size and amount of population we limited every city area only to 10 sq. km. The goal was to compare how the central area of each city being used/occupied by attractions of our interest. Every attraction from feature list was consider as having same importance/weight.

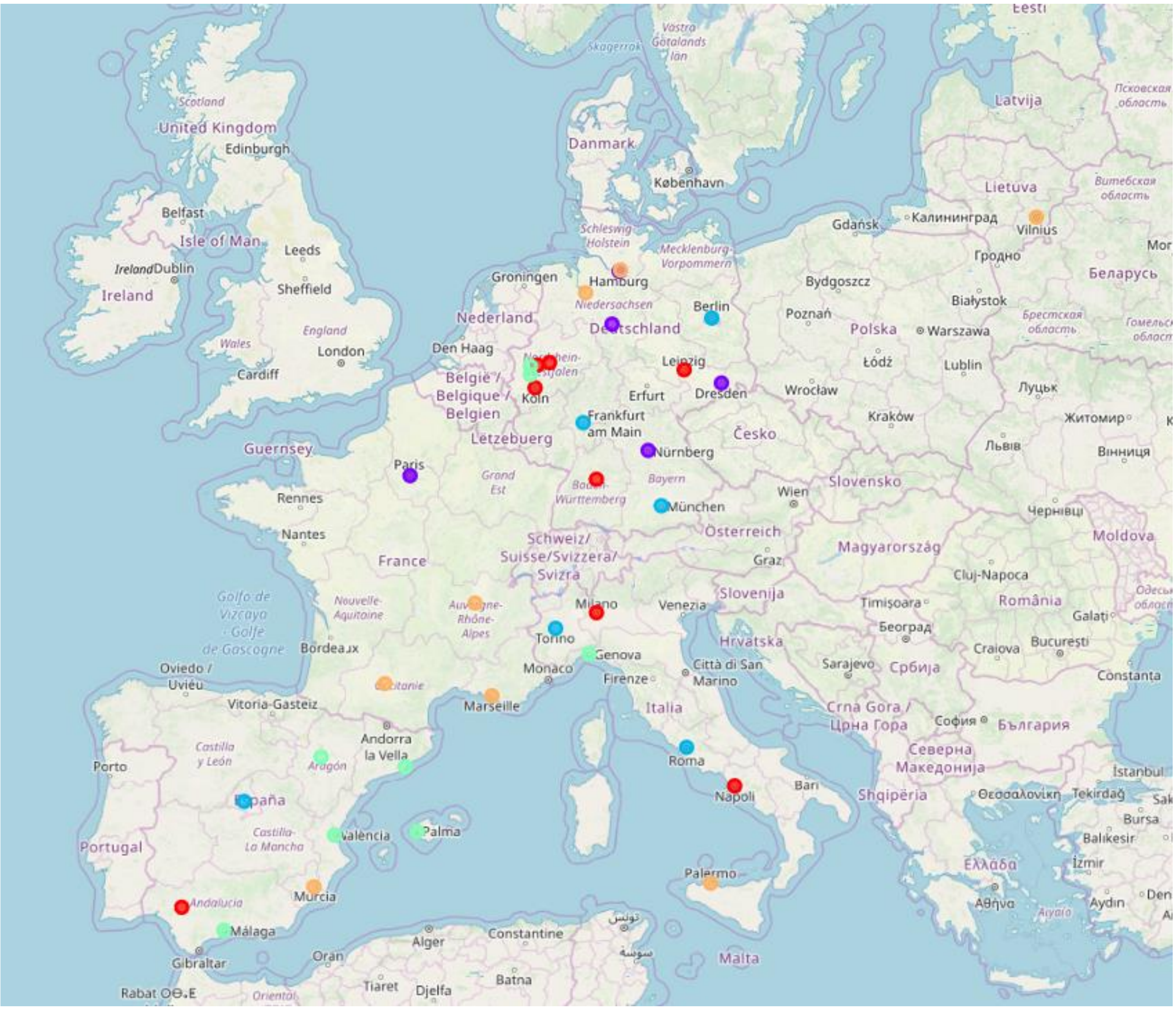
It was notices that some cities don't have some attraction from feature set available in the area. The importance of having at least one attraction from feature set in every city was highlighted by the user. For this reason, model was built to reward every city if it has at least one attraction of every kind from feature list. This adjustment was based on subjective preference of the user and it ensures that the cities with zero attractions from feature list are less likely to be named as sister city.

4. Data clustering

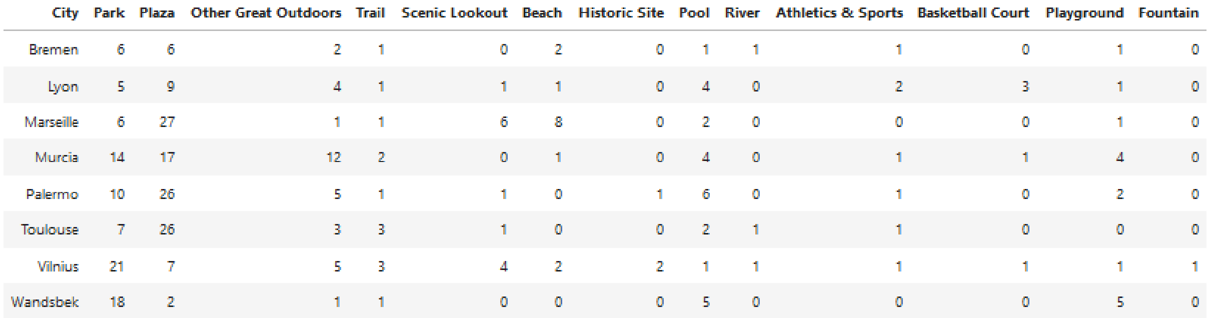
The data was clustered using unsupervised clustering methods.

4.1 K-Means clustering

European Cities were divided to 5 clusters. The results are showed in the map below:



List of cities in Vilnius Cluster with number of available attractions is showed below:



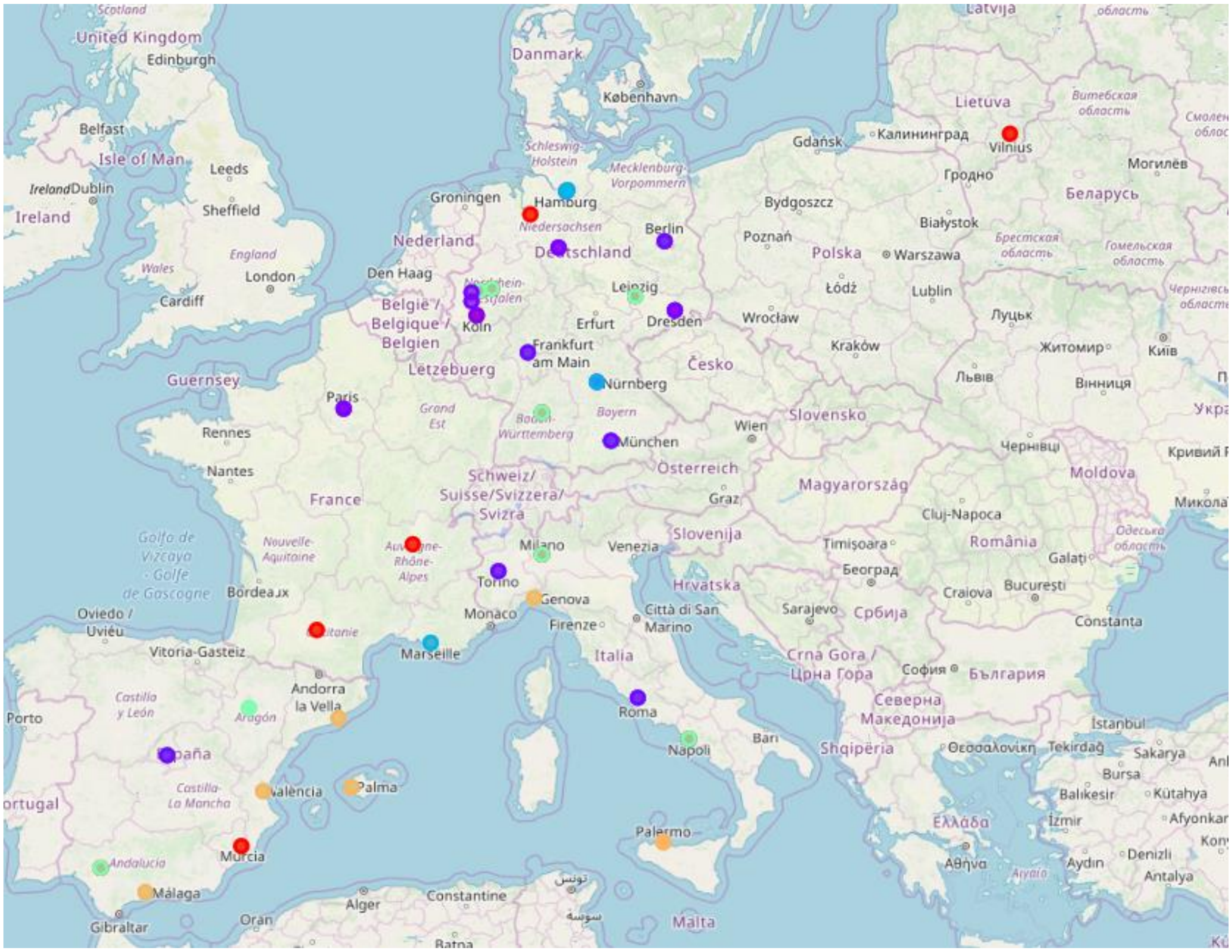
City	Park	Plaza	Other Great Outdoors	Trail	Scenic Lookout	Beach	Historic Site	Pool	River	Athletics & Sports	Basketball Court	Playground	Fountain
Bremen	6	6	2	1	0	2	0	1	1	1	0	1	0
Lyon	5	9	4	1	1	1	0	4	0	2	3	1	0
Marseille	6	27	1	1	6	8	0	2	0	0	0	1	0
Murcia	14	17	12	2	0	1	0	4	0	1	1	4	0
Palermo	10	26	5	1	1	0	1	6	0	1	0	2	0
Toulouse	7	26	3	3	1	0	0	2	1	1	0	0	0
Vilnius	21	7	5	3	4	2	2	1	1	1	1	1	1
Wandsbek	18	2	1	1	0	0	0	5	0	0	0	5	0

The are 7 other cities but Vilnius in the cluster and though could be considered as twin/sister cities.

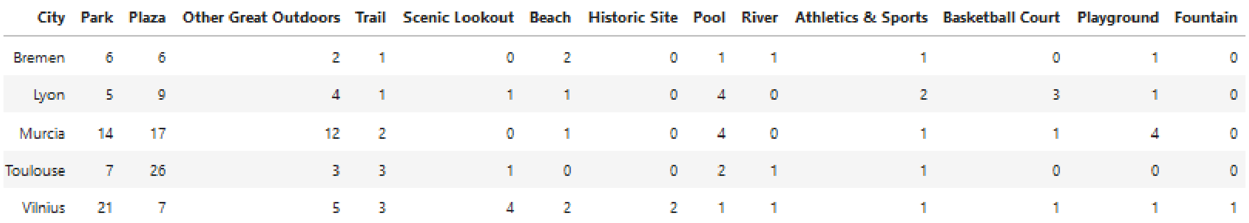
methods.

4.2 Hierarchical - Agglomerative clustering

European Cities were divided to 5 clusters. “Average” linkage was selected, as it minimizes the average of the distances between all observations of pairs of clusters. The results are showed in the map below:



List of cities in Vilnius Cluster with number of available attractions is showed below:



City	Park	Plaza	Other Great Outdoors	Trail	Scenic Lookout	Beach	Historic Site	Pool	River	Athletics & Sports	Basketball Court	Playground	Fountain
Bremen	6	6	2	1	0	2	0	1	1	1	0	1	0
Lyon	5	9	4	1	1	1	0	4	0	2	3	1	0
Murcia	14	17	12	2	0	1	0	4	0	1	1	4	0
Toulouse	7	26	3	3	1	0	0	2	1	1	0	0	0
Vilnius	21	7	5	3	4	2	2	1	1	1	1	1	1

4.3 Results

Both Kmeans and Hierarchical clustering give similar result, however Kmeans assigned more additional cities to the list. We would treat common cities in both clusters as sister/twin cities to Vinius. It is Bremen, Lyon, Murcia and Toulouse. We could see that some cities still have zero attractions of some kinds, but overall the sister cities provides similar signature of attractions.

5. Conclusion

Sister/twin cities to Vilnius were found using clustering methods. As a main criterion in selecting the cities were having at least one attraction of each kind from feature set. From the tabled above we could see some cities still doesn’t meet requirements. We could conclude that either our feature set is too broad, and it is hard to find similar cities to Vilnius, or Foursquare data set is not sufficient and doesn’t provide full list of available attractions in the cities. Most likely the second statement is true.

Even though the work described in this paper provide good base for finding sister/twin cities based on feature set, it is very important to understand that availability of true data on cities and facilities is present. Model or methodology can be perfect, but if the data is not comprehensive the result can be poor.

It is very important to build comprehensive and complete data in order to achieve best result driven by data since. Additional API such as Google Places could be used to improve the result.