World Cities Clustering Based on Places of Interest

1. Introduction

Problem

Imagine for example that we work in a travel agency, and are trying to come up with special promotions for returning customers. How to come up with special offers that may target cities that travelers are likely to enjoy? Or if you are a traveler and are planning next cities to visit around the world?

Approach

The goal is to cluster a large number of cities, based on the points of interest that cities offer to their visitors. And the main intuition behind this approach is that if a traveler enjoyed a particular city (or group of cities) the chance that he or she will enjoy cities that offer similar points of interest is higher than otherwise. For example, if you enjoy nightlife you might be more interested in visiting cities that offer wide varieties of bars and night clubs, or if usually are more fond of cultural and heritage, cities that tend to have interesting museums and other cultural venues.

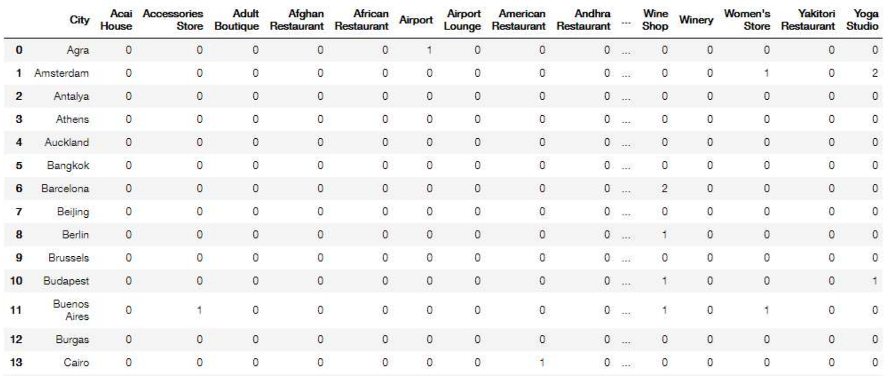
The algorithm envisaged to groups cities with similar points of interest is k-means, a clustering unsupervised machine learning algorithm.

2. Data

We collect information of relevant cities from the top 100 cities most visited in the world in 2015, as described in the *Euromonitor International* report. From this report, we extract the cities, and corresponding countries. Then we find the city geographical location, by discovering each city latitude and longitude.

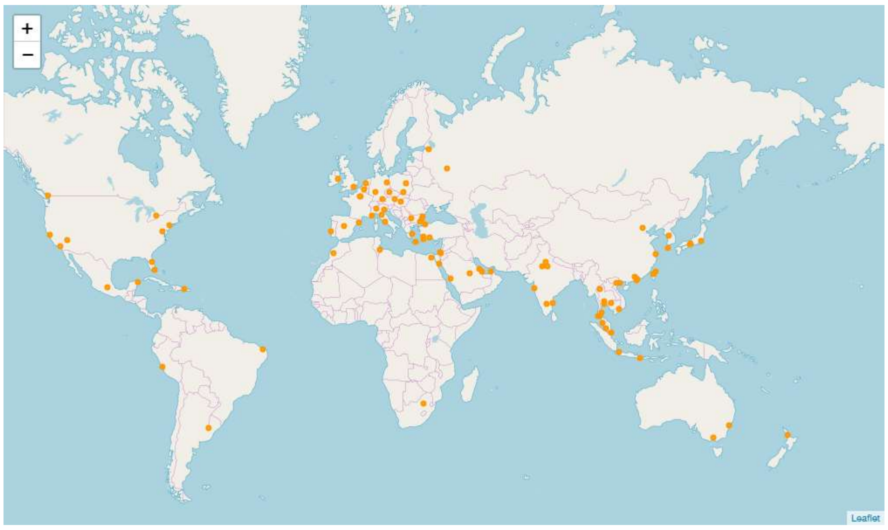
And finally, we query the Foursquare API, available from https://developer.foursquare.com/, to gather information about the venues available in each city. We start by gathering the venues that are in the radius of the city geographical location, and aggregate all the venues by city, by adding the venues for each category for each city.

The final dataset includes a list of cities, each city latitude and longitude, and a vector of categories sums that represent the points of interest available in the city. The following figure illustrates the first rows of the dataset, a vector of categories sums for each city:



	City	Acai House	Accessories Store	Adult Boutique	Afghan Restaurant	African Restaurant	Airport	Airport Lounge	American Restaurant	Andhra Restaurant	...	Wine Shop	Winery	Women's Store	Yakitori Restaurant	Yoga Studio
0	Agra	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0
1	Amsterdam	0	0	0	0	0	0	0	0	0	...	0	0	1	0	2
2	Antalya	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
3	Athens	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
4	Auckland	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
5	Bangkok	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
6	Barcelona	0	0	0	0	0	0	0	0	0	...	2	0	0	0	0
7	Beijing	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
8	Berlin	0	0	0	0	0	0	0	0	0	...	1	0	0	0	0
9	Brussels	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
10	Budapest	0	0	0	0	0	0	0	0	0	...	1	0	0	0	1
11	Buenos Aires	0	1	0	0	0	0	0	0	0	...	1	0	1	0	0
12	Burgas	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
13	Cairo	0	0	0	0	0	0	0	1	0	...	0	0	0	0	0

And, the following figure includes a world map illustrating the cities available in the dataset generated using Folium, more information available from https://python-visualization.github.io/folium/:



3. Methodology

In a nutshell the goal is to cluster cities based on points of interest (venues) retrieved using the Foursquare API, more information about the API is available from https://developer.foursquare.com/.

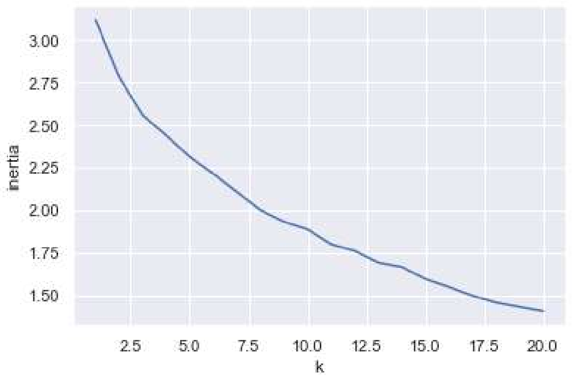
The following steps were performed during the analysis:

1. Pre-process data and create an initial dataset of cities including latitude and longitude:
    i. For each city in the list of interest retrieve geographical information, namely the city latitude and longitude;
2. Retrieve information from the FourSquare API:
    i. For each city, retrieve the venues from Foursquare API, that are centered in the city's latitude and longitude in a 50 km radius;
    ii. Each venue has the category information (e.g. *Hotel*, *Winery*), one-hot encode the venue category for every venue and add the city name information;
    iii. Next group together by city using the sum for every category;
    iv. The result is a dataframe that contains, for each city, a vector representing the venues of interest found in the city;
3. Run the clustering algorithm:
    i. Run k-means, using k number of clusters between the range 1 and 20, and compute the inertia (a measure of the cluster quality) in each iteration;
    ii. Plot the number of clusters *versus* inertia to find an appropriate number of clusters using the *elbow method* approach;
    iii. Since there is no clear *elbow* in the previous plot, i.e. no clear steepness change in the line, we choose to use 8 clusters;
    iv. Build the final model by running k-means with k=8;
4. Explore the results.

4. Results

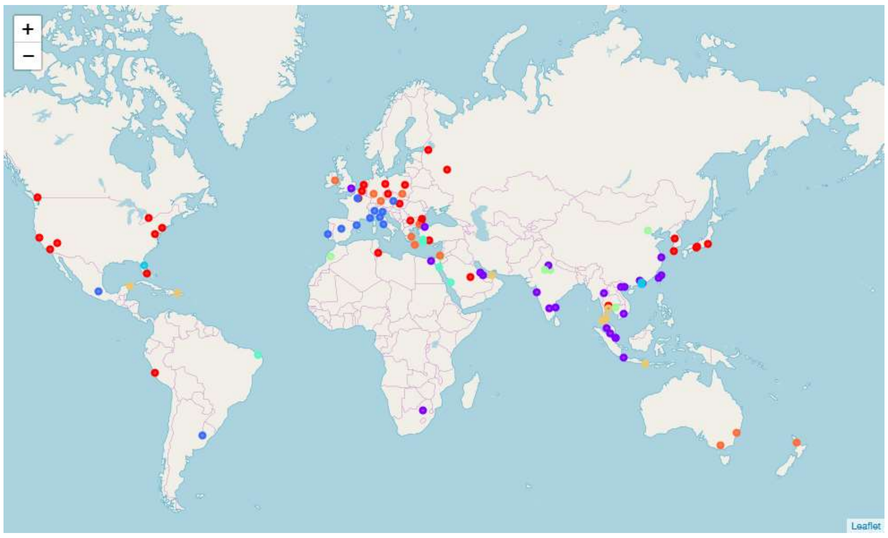
K-means algorithm requires the number of clusters to infer to be defined *a priori*. To try to come up with a sensible number of clusters to use, we refer to a common technique called the *elbow method*. In a nutshell, we run the algorithm several times, using a different number of clusters *k*, and for each iteration compute the result inertia, a common measure of the *quality* of the final set of clusters.

The following plot illustrates inertia measures, over different number of *k* clusters.



Usually there is a steep change in the line, that tells us that after that number of clusters the information gain over all clusters is being marginally increased. This does not happen in this plot, but we choose to use 8 clusters, since there seems to be a slight change of steepness around that point.

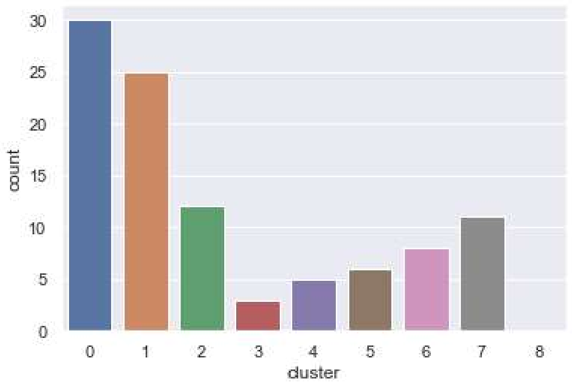
Next we run the k-means algorithm using a pre-defined number of 8 clusters, on the dataset with the vector that contain the average of venues categories. The following figure illustrates the final results assignments, since each city marker is colored by cluster, i.e. same color means same cluster.



5. Discussion

We used k-means to cluster cities all around the world based of information about venues centered around these cities. By a visual analysis of the world map representing the cities colored by clusters, it seems that the results are not very off. Many European and North American cities were grouped together, which makes sense when comparing with groups that include for example Cities on Arab countries, or in the Asian continent. The venues of interest available in these different countries is maybe explained by the cultural and heritage differences. Another example is the cluster that contains the cities in Southern Europe.

The *elbow method* used to try to come up with a sensible number of clusters to use did not yield a clear result, there is no clear steepness change in the curve. We choose to use 8 clusters because it seems there is subtle change in the tendency of the line around that area. But given that there are some clusters with a very small number of members may the number of clusters should be reduced. The following plot illustrates the number of cities in each cluster:



6. Conclusion

The initial goal was to come up with a strategy (approach) to create informed promotions for returning customers in a travel agency. These promotions would like to target cities that travelers would probably enjoy based on other cities they had already visited. The main idea was to group cities that offer similar venues of interest to visitors, and the reasoning behind was that if a person enjoyed a city there is a good chance that he or she would also enjoy other cities that offer similar venues of interest.

To create the groups of cities we based our clustering algorithm on vectors representing the venues of interest available in a 50 km radius of the city center, defined using a latitude and a longitude. Based on the *elbow method* approach we choose to use 8 clusters, a number that could possibly be reduced.

The final result of the analysis is a set of clusters of cities, that can be used to create target promotions for returning customers in a travel agency. We can look at previous cities a person visited, and if this, or a very close, city is available in any cluster, pick other cities in the same cluster to create a special (targeted) promotion for that person. The final set of clusters, illustrated on a world map is available from https://nunorc.github.io/IBM-DS_Coursera_Capstone/.

Future work includes adding more cities to the initial dataset, coming up with other ways to represent the points of interest, and include more sources of information concerning the venues available in the cities.