Career Management by Hilton using locations cluster

1. Introduction:

Global companies want to keep consistence in the company culture despite their many locations. In order to achieve this goal, the employees relocate many times during their career.
Human Resources manager have the task to ensure the mobility. They want to make sure that the employees feel good in the new city.
I take for the case the example of the company Hilton, and the specific case of teams in UK waiting for their next career step in Europe.

The project is to cluster the different cities where Hilton is implanted, in order to assist the career management. Employees will enjoy the new city, if it has the same structure as the city they liked. It means that Employees will enjoy the new city, if it belongs to the same cluster as the city they liked. Due to the Foursquare limitation, we will focus on Western Europe.

2. Data:

Wikipedia has a list of the Hilton properties worldwide:
https://en.wikipedia.org/wiki/List_of_properties_of_Hilton_Worldwide
The first step will be to extract this chart and make a dataframe.
In order to limit the analysis to the European countries, I will use another Dataset from Wikipedia:
https://en.wikipedia.org/wiki/List_of_European_countries_by_area

Then with Geopy we can get the latitude and longitude of each city, and add it to the dataframe.
Then I will use Foursquare to get the venues in the cities.
The cities will finally be clustered using the frequency of occurrence of each venue's category.

3. Methodology

Step 1: get the data from Wikitable and process them to get a dataframe
In order to import the tables and convert them into dataframes, I used the library wikitables and json.

The Dataframes has 1.003 rows, corresponding to the 1.003 implantations of Hilton Worldwide. I droped the unnecessary columns in the dataframe.

Step 2: get the list of European countries

A similar method is used to import the list of European countries. As there are only 51 countries mentioned, I can check them one by one.
By reviewing the datas, I figured out that some names of the countries were having a '*', corresponding to the notes in the original dataframe. The list of countries is the key to be able to filter the datas of step 1. It needs to be clean.

I used a loop to check the unexpected characters and clean them

Step 3: Filter the locations and make them unique¶

To get a list of implantations only in Europe, I create a new dataframe. The columns fits to the dataframe from step 1. With a loop, I check for each if the country is mentioned in the list of European countries and append it.
The method value_counts is then used to check the number of hotels per city. This method provides not only the list of distinct cities where Hilton is implanted, but also the quantity of hotels in each city. It is thus possible to look for mistakes

At the end of step 3, we have a list of cities in Europe where Hilton is implanted: 230 cities.

Step 4: use Geopy and add Latitude and Longitude to each city

The library Geopy is used to add the latitude and the longitude of each city. Unfortunately, the method stops providing the information after 17 iterations. The solution is to run the code 14 times, to cover all the 230 locations.
The data is first checked by sorting the ascending latitude and focusing on the last rows. For some city, Geopy returned no data. The 6 cities are corrected manually: the latitude and the longitude are checked on google.

The data is checked another time by plotting the implantations on a folium map. Indeed, some cities in the UK and in the US have the same name. In the plot, some false data is identified as well in Asia.

The positions are corrected manually.

The list of locations including latitude and longitude is then available. The characterisation and analysis can start.

Step 5: Characterisation of the cities with Foursquare's venues

For each city, I use the API from Foursquare to get the list of venues near the city center. The list of venues make a characterisation of the city: what is available in the city and to which extend.

The API Foursquare provides json files. The data is processed in to order to get the category of the venue.

The first city (Aachen, Germany) is explored in details. The Foursquare API returns 78 venues.

We create a loop in order to get the venues for each city with the same process.

The data is then processed to quantify each venue's category: first I use the method get dummy, then I take the average value for each venue's category over the venues per city.

The data is then available for cluster: each row represents a city, and each column the average presence of a venue's category. We have 449 different categories.

Step 6: Cluster of the cities using k-Means

Using the library sklearn, and k-Means, I proceed with a cluster.

The number of cluster is set to 6, and the iterations to 12.

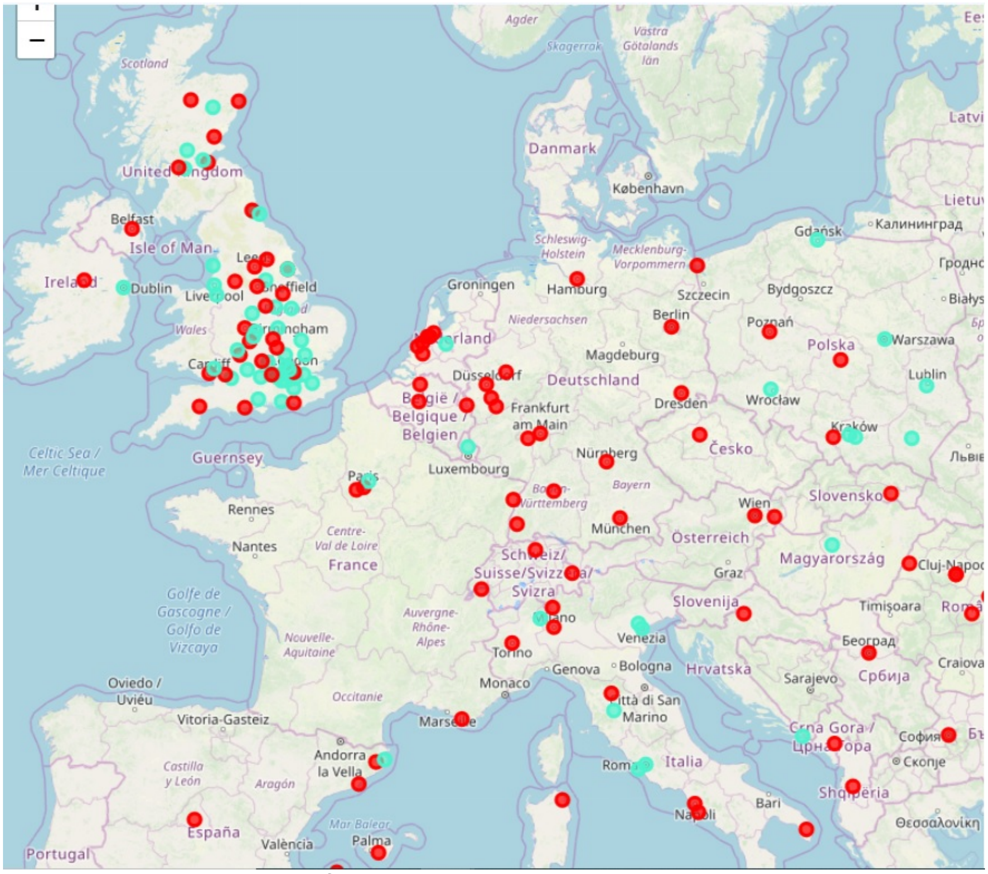
The Cluster labels are then put back in the dataframe.

Step 7: Represent the clusters on the map:

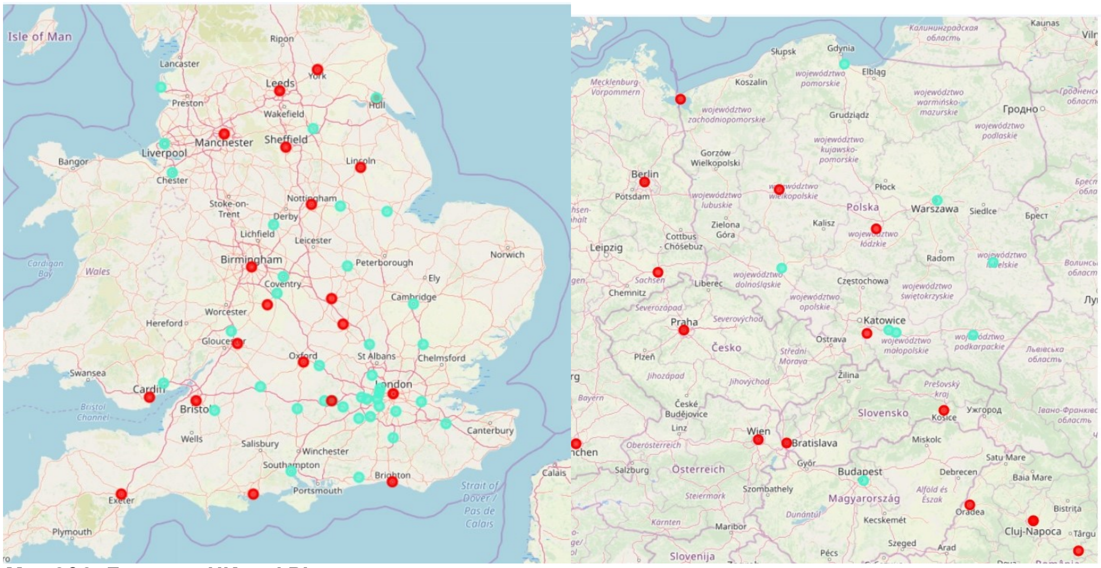
The results are plot on a folium map, where each location is marked with a color corresponding to its cluster.

4. Results

On the map, we observe that many locations on smaller cities in the UK belong to the same cluster as cities in Poland.
For our target to identify options to be studied by the HR manager. We can assume that for employees satisfied with the situation in the UK, they can consider going in Poland where the characteristics of the cities are similar.



Map1: Locations and clusters of Hilton in Europe



Map 2&3: Focus on UK and PL

5. Discussion

Several limitations need to be highlighted in the simulation:

- Parameters on Foursquare, and limits of the API:

Due to the quotas on Foursquare, we limit the venues to a radius of 500 m around the City Center and 100 venues. It is not to have an overview of a city. By enlarging the limit of venues or Furthermore, for some cities no venue has been found. These cities have been put together in a single cluster.

- Parameters on Foursquare: number of categories

Foursquare provides 449 different venue's category. The model considers them as equivalent. In reality, some categories may be considered as more important than others. A further development to consider is definitely to reduce the global number of venues by making groups of categories. The next option would be to weight the most important ones after asking the employees to quote which parameters they consider as the most important.

- K-means clustering

The parameters of the clustering have been defined after a few iterations. I could find out that the repartition between the clusters remains stable: the 2 first clusters remain significantly higher than the rest. This situation could be proven by the plot of the "elbow curve", representing the squared error as a function of k, the number of clusters.

6. Conclusion

The initial problem was to provide some additional inputs in order to support decision making process. For the considered specific population (UK), the model answers the question. For a global use through all locations, a deeper use of Foursquare is required.