# Looking for a flat in Paris with data science

Introduction

Living in such a multi-cultural and deeply historical city as Paris is extremely interesting, but it can be challenging to find your perfect place to live.

I am now renting an apartment in a lovely Parisian suburb, but looking for a place of my own. I would love to stay in the same neighbourhood, but as the offer is not that big, I have to consider other options as well. I want to find a perfect area resembling in the lifestyle the one I'm living in now.

To do so, I used the Foursquare data and clustered the neighbourhoods taking into account the available venues and activities. Intuitively simple, this task uncovered several challenges related to clustering categorical multidimensional data. This post is about these challenges and my discoveries about life in Paris districts.

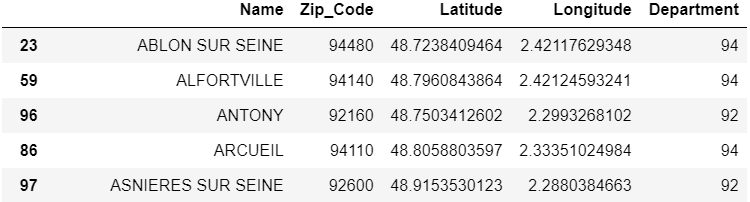
Data

I used two sources of data:

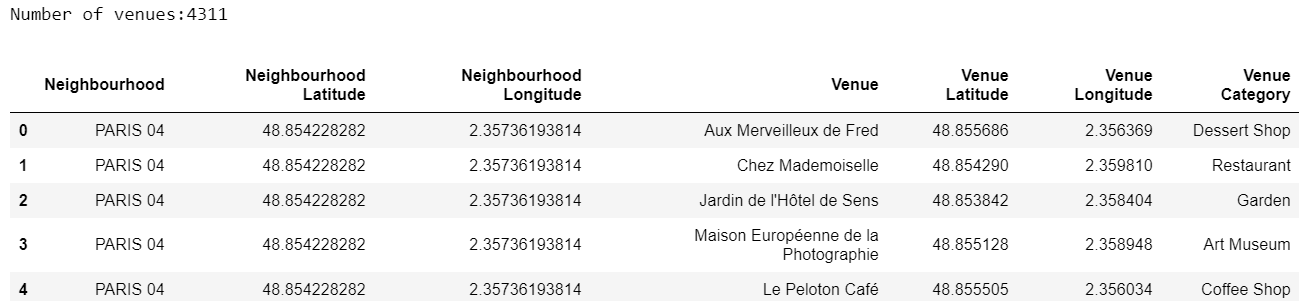
* Information about Paris neighbourhoods containing their names, zip codes and geographic coordinates (in open access at the French Government official web site: <https://www.data.gouv.fr/fr/datasets/base-officielle-des-codes-postaux>/);
* Information about the venues from the Foursquare API open for developers: [https://foursquare.com/developers/](https://foursquare.com/developers/apps/LJZO500UG2WRLMIB3X3TAJLLZVKQURND3DXC3IARUPROJDZF/settings).

The data preprocessing consisted of:

* downloading the zip codes of France and preprocessing them using *pandas* library to leave only Paris and its closest suburbs area also called "Petite Couronne";
* connecting the boroughs zip codes to the GPS coordinates of their centres;



* connecting to the Foursquare database and getting the information about the venues for each borough or neighbouring town;
* setting the limits for the previous point search to one hundred venues and a one-kilometre distance from the centre of the district;
* store the resulting query in a *pandas DataFrame*.



Once the venues extracted for each borough in Petite Couronne, I realised that some of the neighbouring towns, especially the smallest and the furthest from Paris itself, only have a couple of venues:

A close up of a piece of paper

Description automatically generated

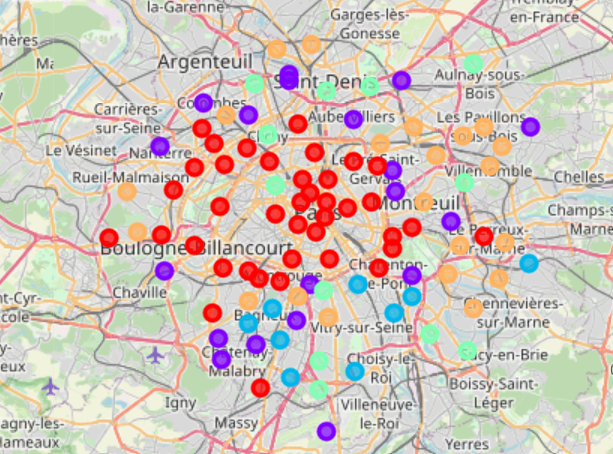
I live in a quite lively neighbourhood, so I directly discarded all the areas possessing less than ten sites. This left me with 3998 distinct venues distributed through 99 boroughs.

Methodology

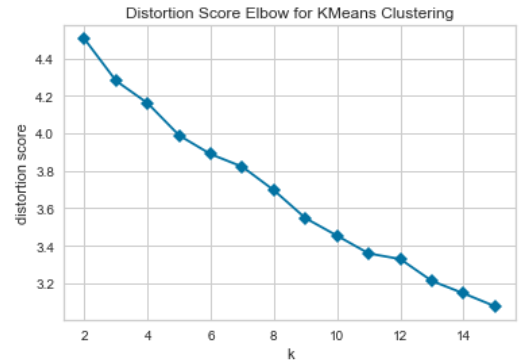
The first method that I tried was inspired by the IBM Data Science Professional Certificate training. I attempted to cluster the remaining boroughs with respect to the proportion of venues of different types. For example, for the town I currently live in, 10% of its sites are French restaurants, then go pizza places, Japanese restaurants, parks and pubs all having 5% and so on.  This approach corresponds to the following representation of districts as categorical vectors:



I used the *scikit-learn* ***K-Means*** unsupervised clustering algorithm to identify my clusters; to start on with, I specified that I want to get 5 clusters. The algorithm converged, and here is the result I got, mapped using ***folium*** library for plotting geospatial data:



Then I decided to find the optimal number of clusters. One can do it h the "elbow method" ([link](https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/)). But here a surprise was waiting for me: there was no elbow! The distortion score decays monotonously, and the algorithm is unable to find an elbow point (I used the ***KElbowVisualizer***functionfrom the *yellowbrick*library).



After carefully going through the data, I found out that a large number of districts got assigned to the same cluster, despite having nothing in common. It is a kind of cluster “everything else”:



This happened due to the categorical feature representation. The absence of the elbow and the mega cluster indicate that the selected representation is not optimal for the data.

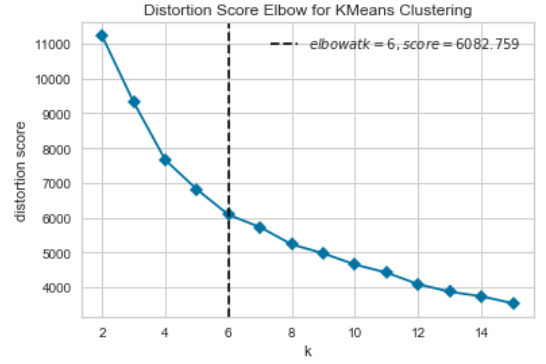
Furthermore, after scrutinising the data, I found out that the number of distinct venue categories in my dataset is 317, which might be excessive for my dataset size. So I decided to create bigger super-categories, for example, Chinese, Japanese, Korean and Thai restaurants all go to the "Asian Restaurant" super-category.

On top of that, such proportional features do not take into account the total number of venues in each borough; however, when looking for a new place to live, this seems to be an essential parameter to take into consideration. I thus propose to use a raw number of venues of each category instead of a proportion.

To sum up, the three main ameliorations are:

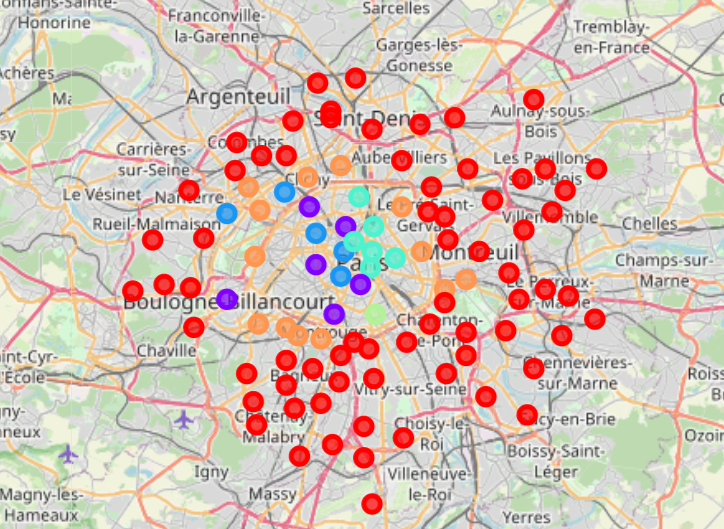
* Replace categorical representation of feature vectors by a continuous one;
* Apply feature engineering – create super-categories;
* Take into account the total number of venues by replacing the proportional features by raw quantities.

This time, KElbowVisualizer does not encounter any problem finding an optimal number of clusters.

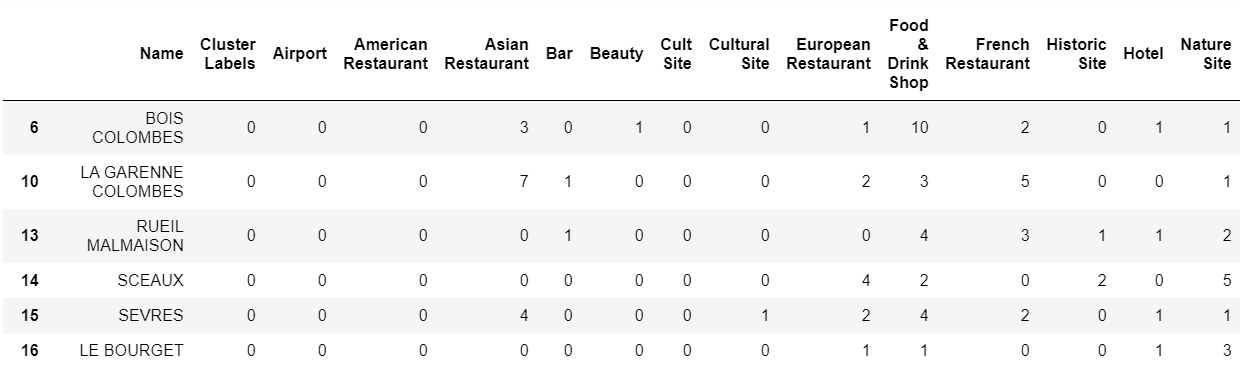


Results

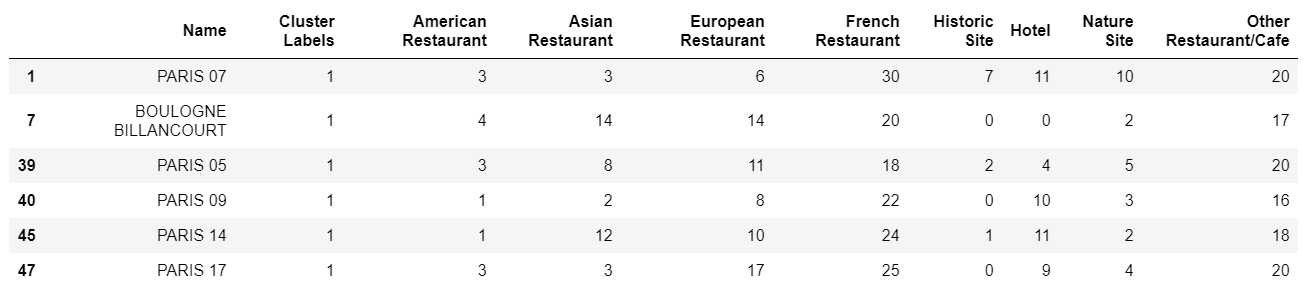
The final clustering results are presented below. The new clusters appear to be clear and interpretable.



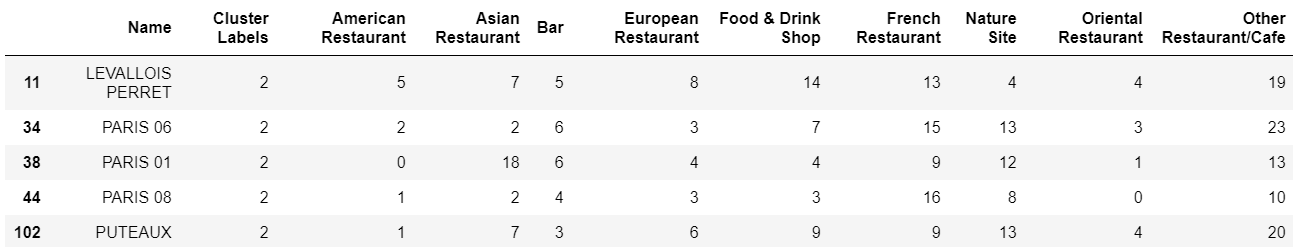
Cluster 0 (red) delineates residential areas with lots of shops and supermarkets but fewer restaurants, bars and cafés:



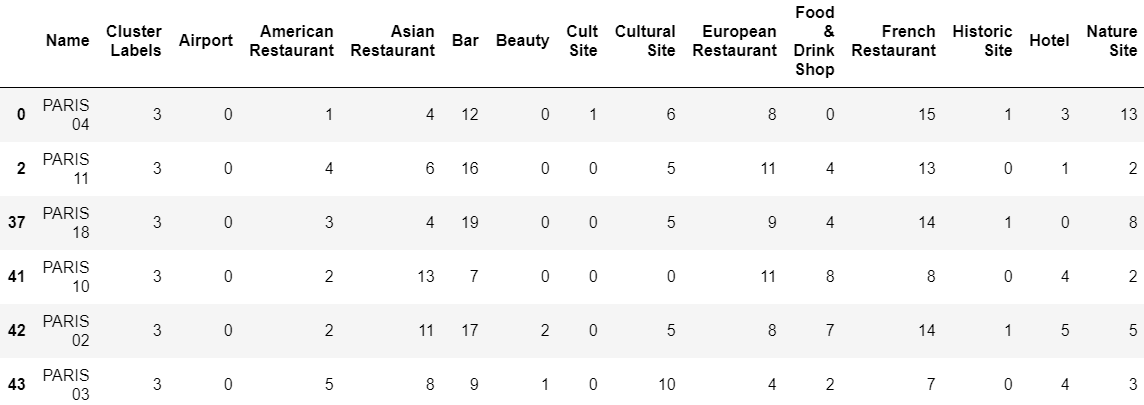
Cluster 1 (purple) is a truly gastronomic cluster: it has lots of French and European restaurants, as well as non-geographic types of restaurants:



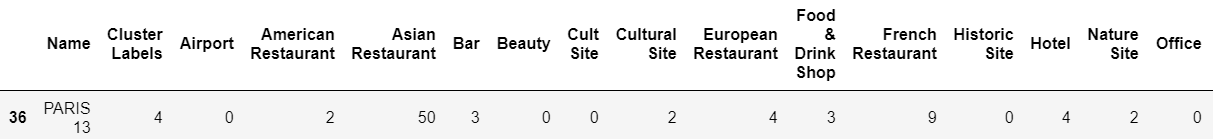
Cluster 2 (blue) is my favourite one! In addition to French and non-geographic restaurants, has a lot of shops and natural sites, such as parks and gardens:



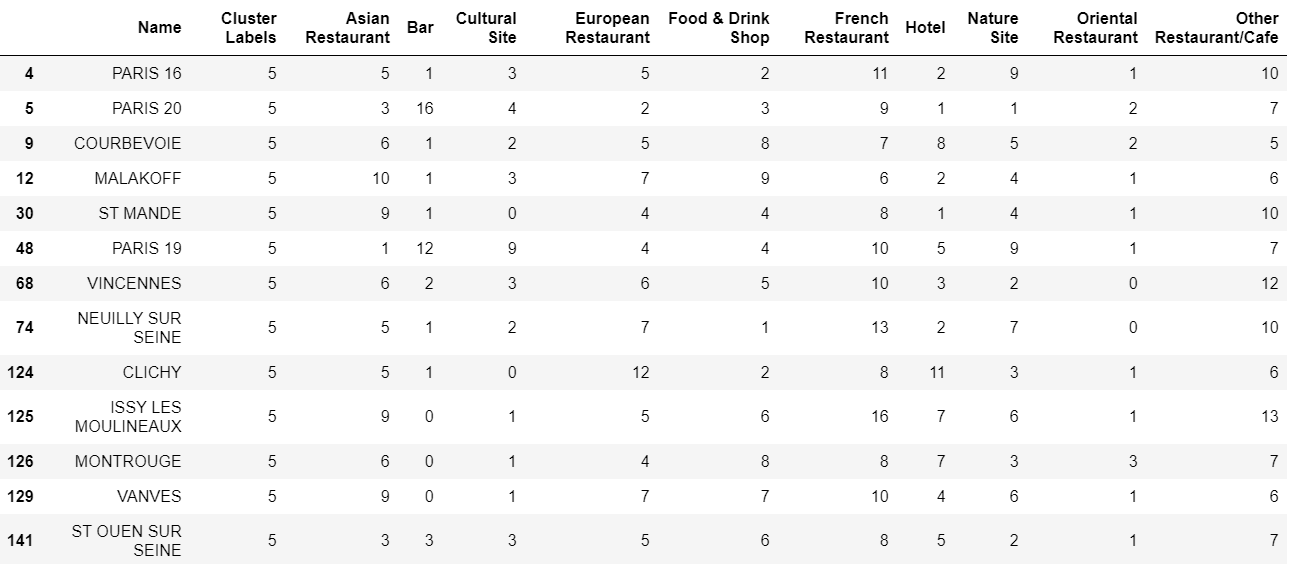
Cluster 3 (green) is a night-life cluster:



Cluster 4 (yellow) only has one district in it, and it is a curious one: it features 50 Asian restaurants:



Cluster 5 (orange) seems to have a bit of every type of venue, such a happy mean. It also geographically delineates the Paris boundary and corresponds to the respectful suburbs:



Discussion & Conclusion

For multidimensional data such as information about districts, it is challenging to find a good numerical representation and evaluate the quality of clustering. As this post demonstrated, checking the “elbow” plot can be used for this purpose, as well as manual analyses of resulting clusters. The main approaches that I used was careful feature engineering (joining categories) and switching from categorical features to continuous ones.

The task of clustering a city’s boroughs is a very vast one, as it can have numerous implementations according to one's specific needs.

I had to discard the first methodology due to the lack of distinction between the different boroughs, and I think the last improved methodology applies to the most significant amount of people looking for a new place to live in Paris. The clustering provides a clear distinction between the different areas and takes into account the level of the liveliness of a borough. On the non-technical side, this study proves once again that whatever are your preferences for a place to live in, you can find your perfect home in the magical city of Paris!