# Looking for a flat in Paris like a data scientist

Introduction

Living in such a multi-cultural and deeply historical city as Paris is extremely interesting, but it can be challenging to find the apartment of your dreams here.

I am now renting an apartment in a lovely Parisian suburb, but looking for a place of my own. There is much reading on housing available on the web, but it is often opinionated and focused on prices per square metre. I decided to look at the problem from a different angle – comparing lifestyle and ambience in different neighbourhoods. How does one quantify lifestyle? One way would be to look at the venues and activities available.

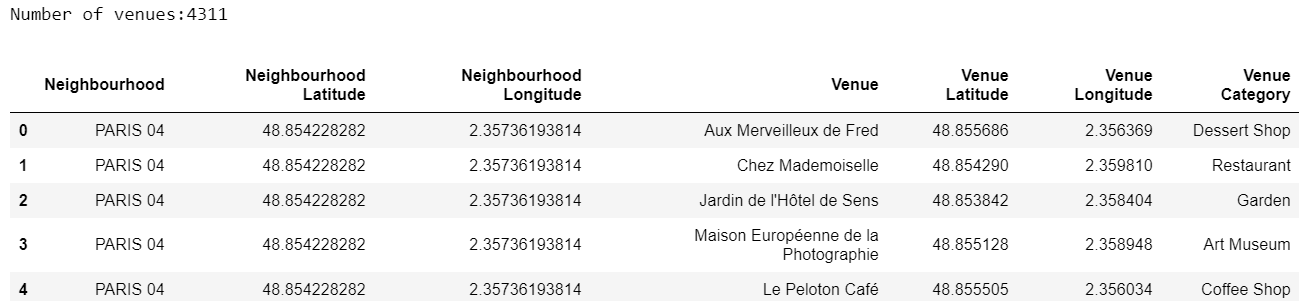
So, I connected to the Foursquare public API, downloaded some data about Paris neighbourhoods and fed them to a clustering algorithm to see if a pattern emerges. Intuitively simple, this task uncovered several challenges related to clustering categorical multidimensional data. This post is about these challenges and my discoveries about life in Paris.

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). I downloaded up to 100 sites per neighbourhood using *requests* python library.

I have done some simple data preprocessing: cleaned the data, joined information about the neighbourhoods with the information about the venues and got a nice and clean *pandas* DataFrame.



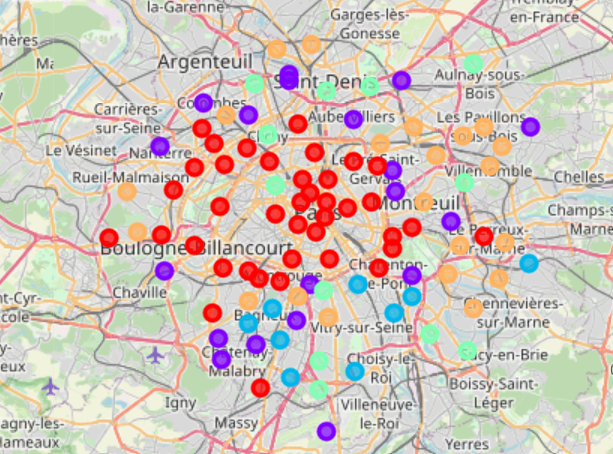
Once the venues extracted for each borough in Petite Couronne (Paris and its closest suburbs), I realised that some of the neighbouring towns, especially the smallest and the furthest from Paris itself, only have a couple of venues. I live in a quite lively neighbourhood, so I directly discarded all the areas possessing less than ten sites. This left me with 3998 distinct venues distributed through 99 boroughs.

Methodology

Having the data prepared, I switched to the central part of the job – clustering. The first method I tried was inspired by the IBM Data Science Professional Certificate training. For each neighbourhood, I ranked the types of venues according to their occurrence frequency. For example, for the district I currently live in, the most significant part of all sites are French restaurants, followed by pizza places, Japanese restaurants, parks, and so on.  This approach corresponds to the following representation of districts as vectors of categorical features:

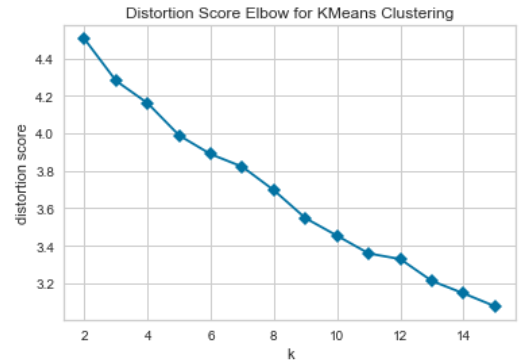


I used the *scikit-learn* ***K-Means*** unsupervised clustering algorithm to identify my clusters; to start on with, I specified that I want to get five clusters. The algorithm converged, and here is the result I got, mapped using *folium* library for plotting geospatial data:



Some structure is visible on the map – clearly, Paris internals are generally different from the suburbs (not surprising, isn't it?). The lack of diversity inside Paris, however, is suspicious. Can I do better by using a different number of clusters? What will be the optimal number? I decided to use the "[elbow method](https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/)" to find it. But here a surprise was waiting for me: there was no elbow! The distortion score decays monotonously, but the algorithm is unable to find an elbow point (I used the ***KElbowVisualizer***functionfrom the *yellowbrick*library):

YellowbrickWarning: No 'knee' or 'elbow' point detected.



Also, after carefully going through the obtained clusters, 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”:



The absence of the elbow and the mega cluster indicate that the selected representation is not optimal for the data. One possible issue can be the categorical feature representation, so I decided to switch to continuous features. On top of that, ranking-based features do not take into account the total number of venues in each borough, which is an essential indicator of the district liveliness. I thus decided to use as a feature the raw number of sites of each type. So my new feature vectors should look like:

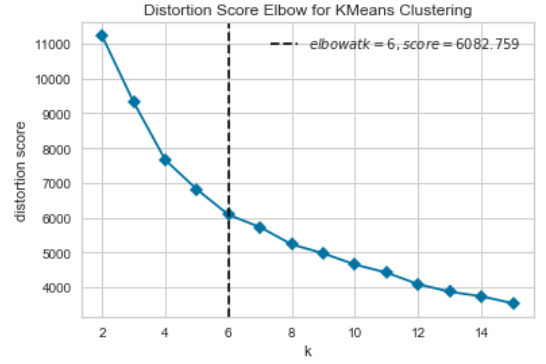
(Count of French Restaurants, Count of Korean Restaurants, Count of Gyms, Count of Parks…)

Next, I found out that the number of distinct venue categories in my dataset is 317, and so is the length of my new feature vector, 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 "Asian Restaurant" super-category.

To sum up, the three main changes are:

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

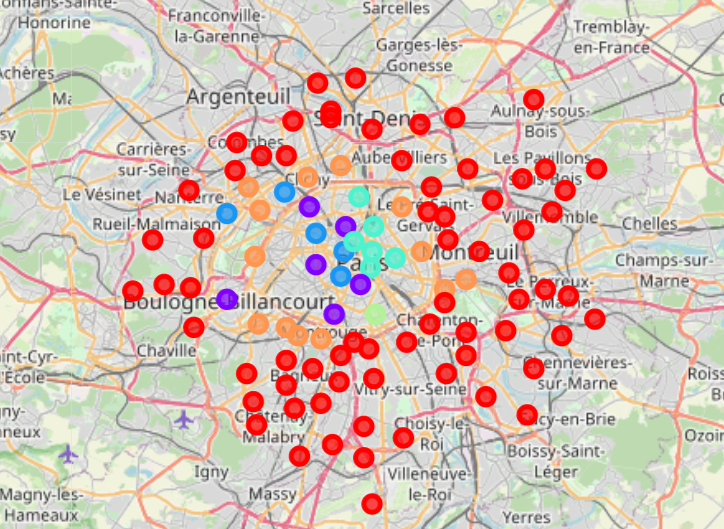
This time, KElbowVisualizer does not encounter any problem finding an optimal number of clusters. The data is clearly more separable into clusters in this representation, and I should divide the neighbourhoods into six clusters.



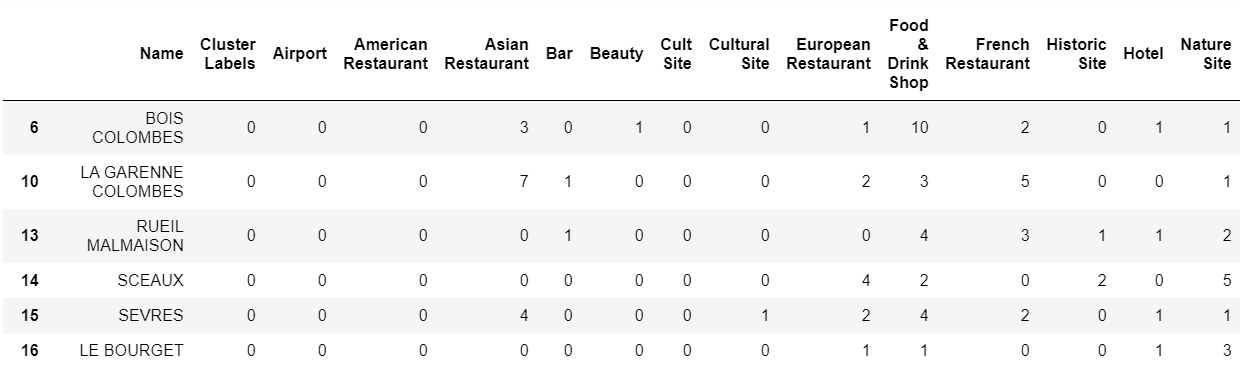
In the next section, you'll see that it actually makes sense!

Results

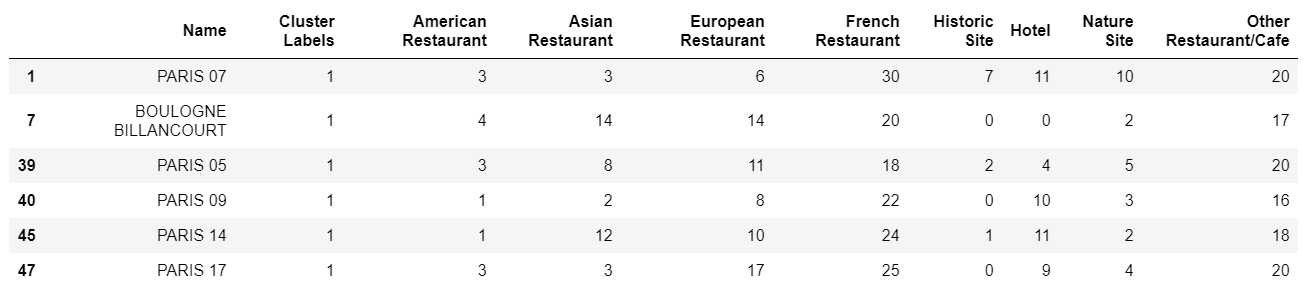
Here is my final result of clustering. Interestingly, now we see some variability inside Paris. Let’s interpret each cluster by looking at the districts inside it.



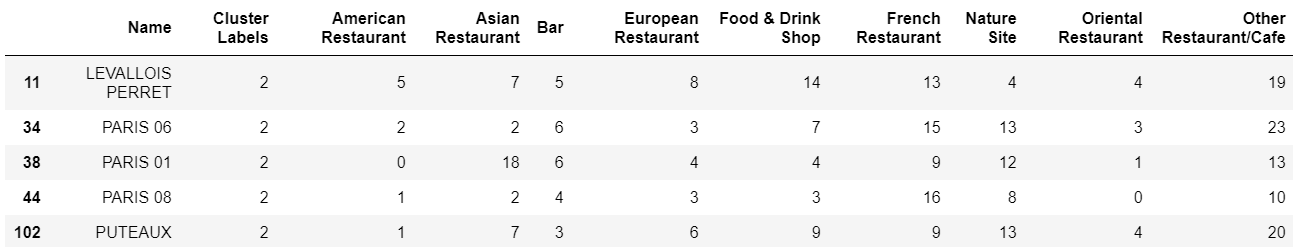
Cluster 0 (red) delineates residential areas with mostly shops and supermarkets but fewer restaurants, bars and cafés:



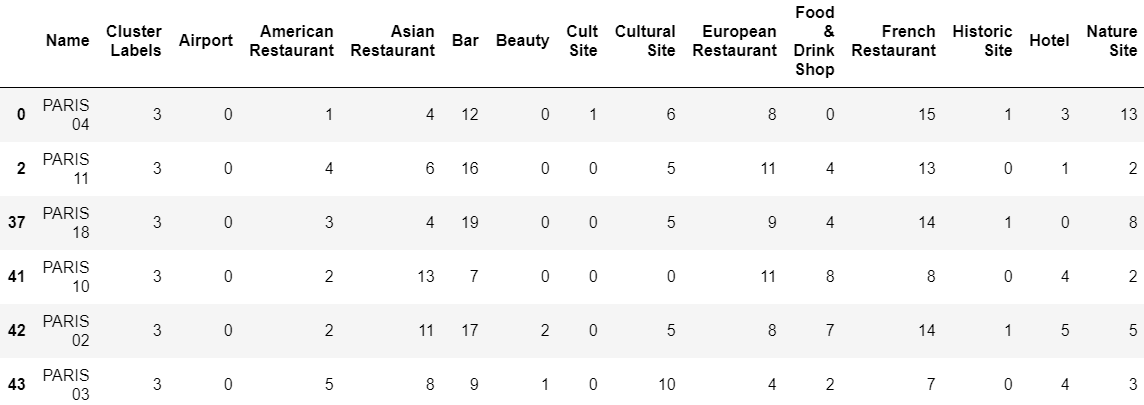
Cluster 1 (purple) is a truly gastronomic cluster: it has lots of French and European restaurants, as well as non-geographic types of restaurants:



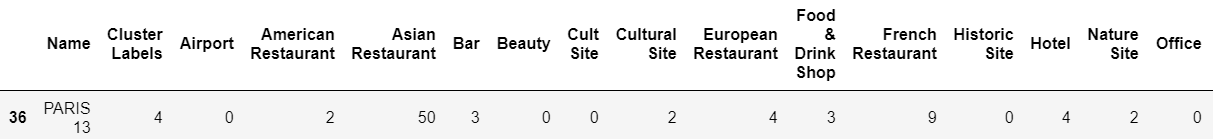
Cluster 2 (blue) is my favourite one! In addition to French and non-geographic restaurants, has a lot of shops and natural sites, such as parks and gardens:



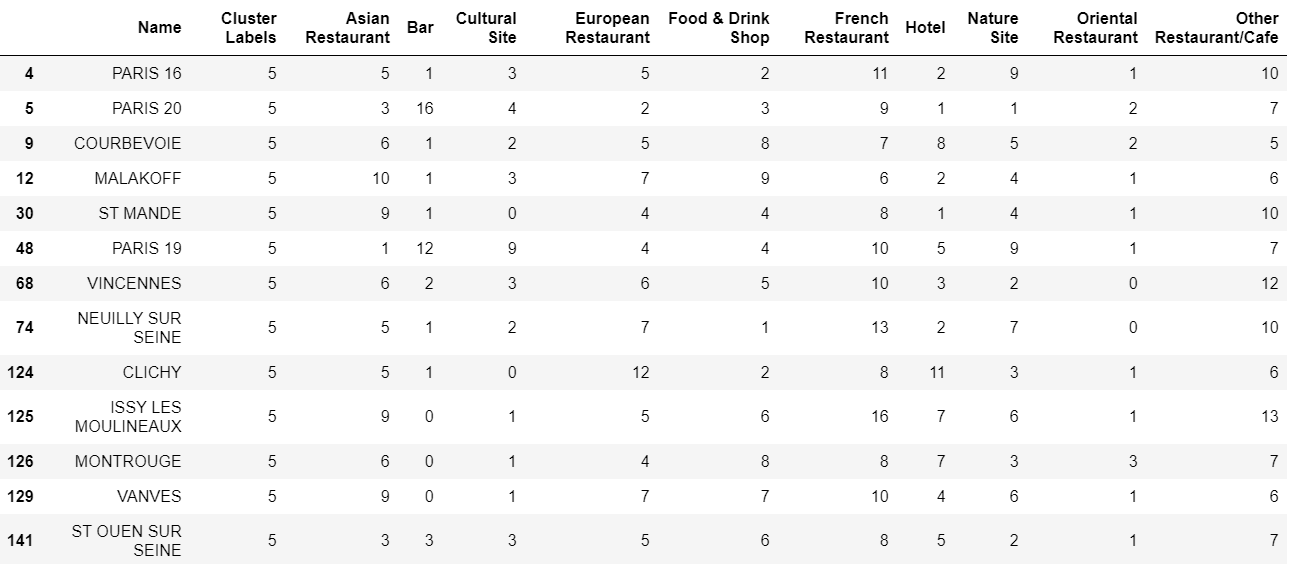
Cluster 3 (green) is a night-life cluster:



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 prestigious suburbs:



Discussion & Conclusion

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

The resulting clustering provides a clear distinction between different areas and takes into account the level of liveliness of each 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!

*Disclaimer*: The conclusions derived in this post should be taken with caution, as I only analysed up to one hundred venues in the inner Paris, while in fact, there are more. Assuming that these 100 were selected uniformly randomly, they can still be considered representative.