Clustering of French Cities

Vianney Mixtur

Introduction

Background

France is divided into 18 regions: 12 in mainland France and 6 elsewhere (1 in Europe (Corsica), 2 in the Caribbean, 1 in South America, and 2 near Africa. We usually talk about Metropolitan France (mainland France + Corsica) and Oversea France (France d'Outre-Mer).

Following the 2016 reform, France is divided as follows :

- 13 Metropolitan regions, including Corsica
- 5 Overseas regions
- 96 departments (Each region is subdivided in one or more department)
- and further subdivisions that we will not talk about here

This what the map of the regions looks like:

Problem Statement:

Before the 2016 reform, the regions were smaller, in fact Metropolitan France went from having 26 to just 13 regions. The debates were intense to know how the new cut would made and which regions would be merged.

The goal of this project would be to explore if by clustering the France cities **we are able to reproduce France regional division**. That is find if we can characterize the regions with common "features" found in the cities that make up the region.

This could be of particular interest for the **French Government** as, 4 years after the reform, it could justify or deny the choices made then.

Data

In order to solve this problem we will use the list of french cities here as well as climatic data from the openweather api. Finally as required, we will use data from the Foursquare API to further categorize our cities.

Cities

The cities data have been exported from the website into a csv file that can be found in the Data folder of the repository of this project. Note that the CSV has no headers. You can find a description of the columns at the original website but find below a summary of what the database contains.

- Various versions of the name of the cities
- Coordinates of teach city (so precious to call the Foursquare API)
- Population data (will certainly be very useful as well)
- Other geographical data (min and max altitudes, area)
- Department number (it would be interesting to check if the algorithm learn and find the relationship between regions and departments)

Weather

We will use the openweather API to retrieve the weather predictions of the day for each city. From this API, we can get data such as minimum recorded, maximum recorded, average minimum, average maximum for the following climatic features :

- Temperature
- Pressure
- Humidity
- Wind
- Precipitation
- Clouds

Venues

We will use the Foursquare API to retrieve the venues that can be found in each city. If you access this notebook from outside this course please check the developer.foursquare.com website. In a nutshell, with this API you can get the following.

- What are the most popular venues are there in each city?
- How many of those are there?
- How many tips (recommendations, opinions from visitors) those venues received?

Methodology

Statistical Analysis, Scoping and Filtering

We used the pandas describe method to get a quick report on the data.

	Name	Department	Postal Code	Population	Density	Area	Longitude	Latitude	Altitude Min	Altitude Max
count	36700	36700.0	36700.0	3.670000e+04	36700.000000	36700.000000	36700.000000	36700.000000	36568.000000	36568.000000
unique	34100	102.0	6082.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
top	sainte colombe	62.0	51300.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
freq	14	895.0	46.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean	NaN	NaN	NaN	1.768011e+03	154.996049	17.257375	2.786424	46.691117	193.156831	391.105694
std	NaN	NaN	NaN	1.475622e+04	704.510109	143.746399	2.966138	5.751918	194.694801	449.308488
min	NaN	NaN	NaN	0.000000e+00	0.000000	0.040000	-62.833300	-63.082900	-5.000000	0.000000
25%	NaN	NaN	NaN	1.940000e+02	18.000000	6.400000	0.700000	45.150000	62.000000	140.000000
50%	NaN	NaN	NaN	4.300000e+02	39.000000	10.755000	2.650000	47.383300	138.000000	236.000000
75%	NaN	NaN	NaN	1.061000e+03	91.000000	18.370000	4.883330	48.833300	253.000000	435.000000
max	NaN	NaN	NaN	2.243833e+06	26660.000000	18360.000000	49.443600	55.697200	1785.000000	4807.000000

We noted a few interesting things from this table:

- The cities names are not unique therefore we cannot use them as-is for indexing
- A department can contain up to 62 cities
- Except for the altitudes we do not have missing values
- The population, density and area data are very spread out, with the standard deviation being superior to the mean
- The area max seems to be an outlier

Following this analysis, we decided to focus on the Metropolitan regions and only on cities with at least 10 000 inhabitants.

We built 3 respective datasets for the cities, the weather and the venues.

The cities dataset was containing the name of the city, its population, density, area and coordinates.

The venues dataset was containing the count of venues per venue category.

The weather dataset was containing the temperature, wind and humidity data.

Data Consolidation

After filtering, the cities dataset was not containing cities with duplicates name. However, the original dataset does.
Therefore, the data were consolidated based on the city name and the city coordinates.

Preprocessing

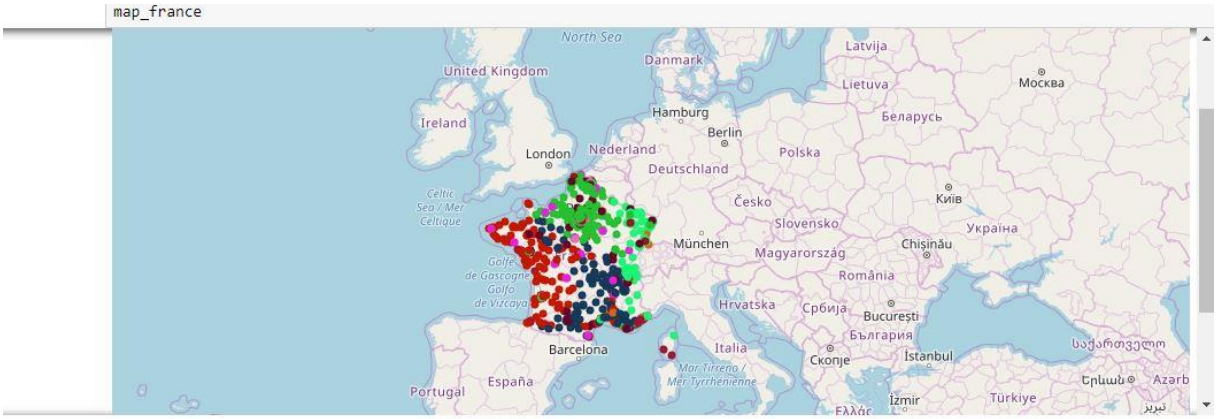
The dataset was then normalized using the *fit_transform()* method of the sklearn.preprocessing library.

Modeling

As we aimed at finding relationships between the cities that match France regional split, we use k-means clustering to cluster the cities.
We set the number of clusters to 13 and randomly initialized the centroids and iterated 50 times.

Results

The results were interesting for a first attempt. As you can see from the map below:



Some clusters are not very represented on this map because only contains a few cities. It is interesting to see, that at least the clustering has split France into 4 main regions:

- The West Coast
- The Northern/Paris region
- Central-South region
- A tiny East Coast region

Discussion

Challenges

This problematic was very challenging and became even more challenging while being worked out.

One of the main challenges, was that we originally planned to use yearly statistical climatic data from the openweather API but those were only available for paid subscriptions.
The second main challenge was also related to this API which does not allow the free users to call the API more than 60 times per minute.

Improvements

Filtering the data as we did in this first quick analysis introduces a bias towards regions with bigger cities. Rather than filtering directly on population, we could have filtered by department so that each region contains approximately the same number of cities.

As previously discussed, we originally planned to use historical statistical weather data to make the model more robust. As it is now, the model can vary on a day to day basis with the weather which is not good.

Last, we standardized the data and therefore made all features equally important.
However, we can assume some features are more important to identify a region. The altitudes from the original city dataset, the number of beaches, forests or water streams from the venues dataset.

Conclusion

This approach to justify or deny a reform is interesting and promising for the future but as of now it cannot really be used in the debates. A different approach such as density-based clustering or classification could produce better results.