Lockdown is almost finished

By William Meunier, May-2020
For Coursera Capstone Project

IBM Developer
SKILLS NETWORK

Table des matières

Purpose of this document

This document is a part of the ***Applied Data Science Capstone by IBM*** certification program on Coursera®. Point of view and hypothesis assess on it are only reflect of the author’s mind.

Introduction

ELEMENT OF CONTEXT

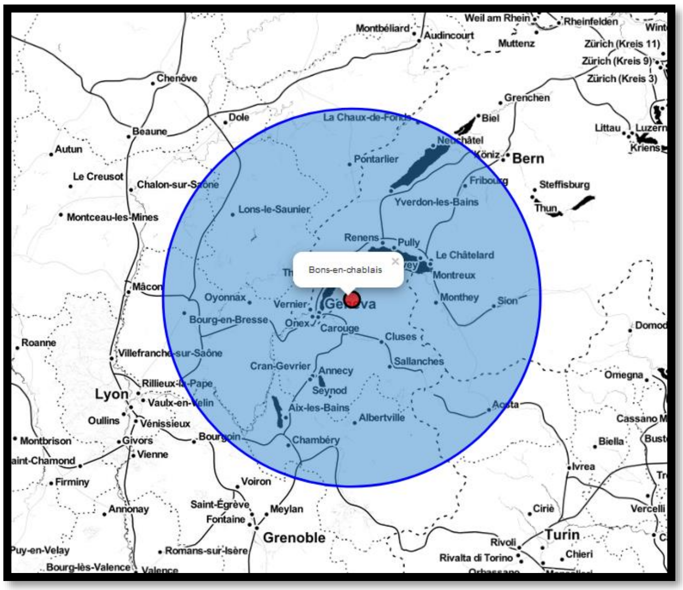
With the Covid-19 pandemic, which began in the beginning of 2020, people, government and world must face of a new way of life. To slow the growth of this virus, many governments decide to put in place lockdown for people. In consequence, many people have to change their daily behaviour and their plan and project for the year.

In France, after almost 50 days of confinement, the life begin to come back at his almost normal level. Nevertheless, the end of lockdown is accompanied with many measures, especially restriction on movement on the national territory, limit to 100 km near the primary location of people and travels are not recommended (almost prohibited). According first estimates, these restrictions could stay in place until the middle of the summer.

In these conditions, many people wonder where they could go during their vacancies.

Major problem is to determine which location could be good for vacancy. 100km, it is not large, but when you leave far away of famous vacancy location or far away of the coast and beach, how determine which location is better than another is?

Currently, some people can be interested in a ranked list of possible destination for their vacancy, in order to gain time or to narrowing the range of possibilities



-

Figure 1: Settings of primary location

.

FIRST SETTING AND HYPOTHESIS

For our study, we set a primary location in ‘*Bons en Chablais’* a small village in France, next to Switzerland.

We choose this location for many reasons. First, it is located in campaign, so it ensure that we do not reach the maximum quota of API used (see after) when we retrieve venues. Then it is near from a Switzerland border, so we can test that our solution works even someone set a location next to a border (as they are closed with Covid-19 restriction)

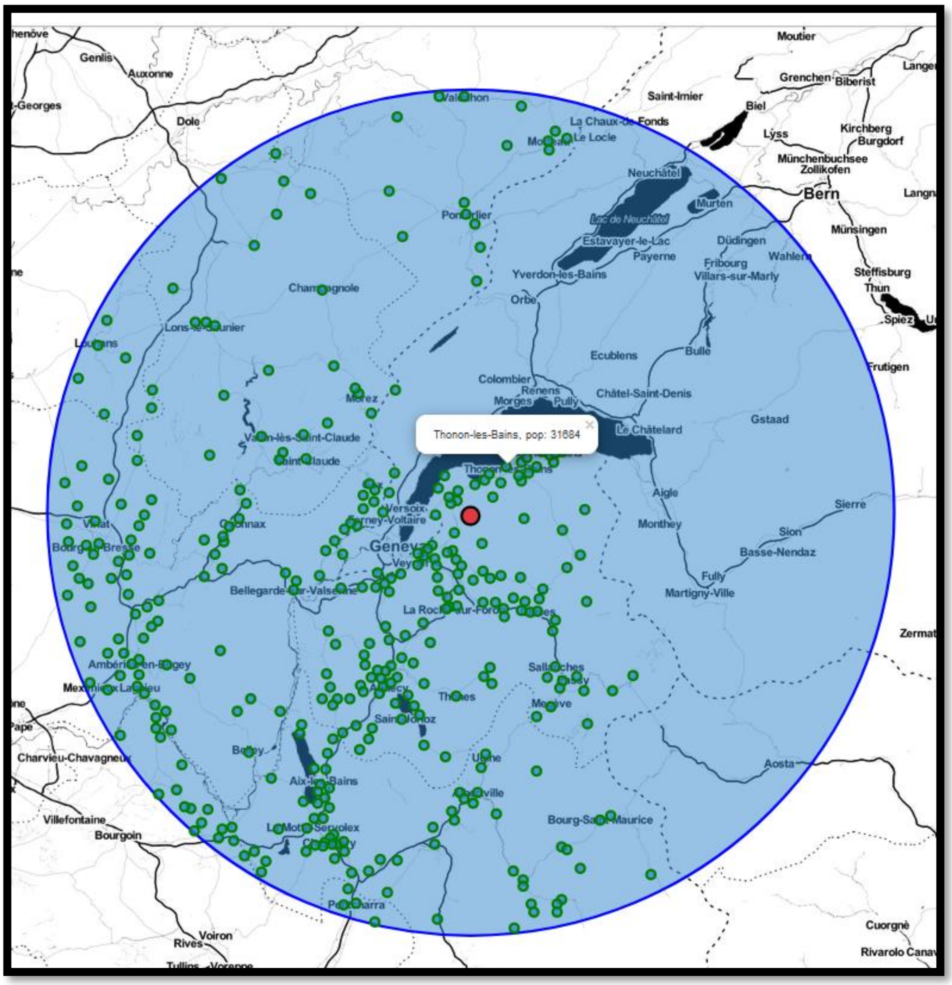


Figure 2: Cities in the Nearby

Overview of the solution

To solve the problematic above, it has been decide to design a small flexible solution. Based on user's input, a location of interest and rank in the choice, a list of city should be provided to help decision.

The workflow for it is presented below.

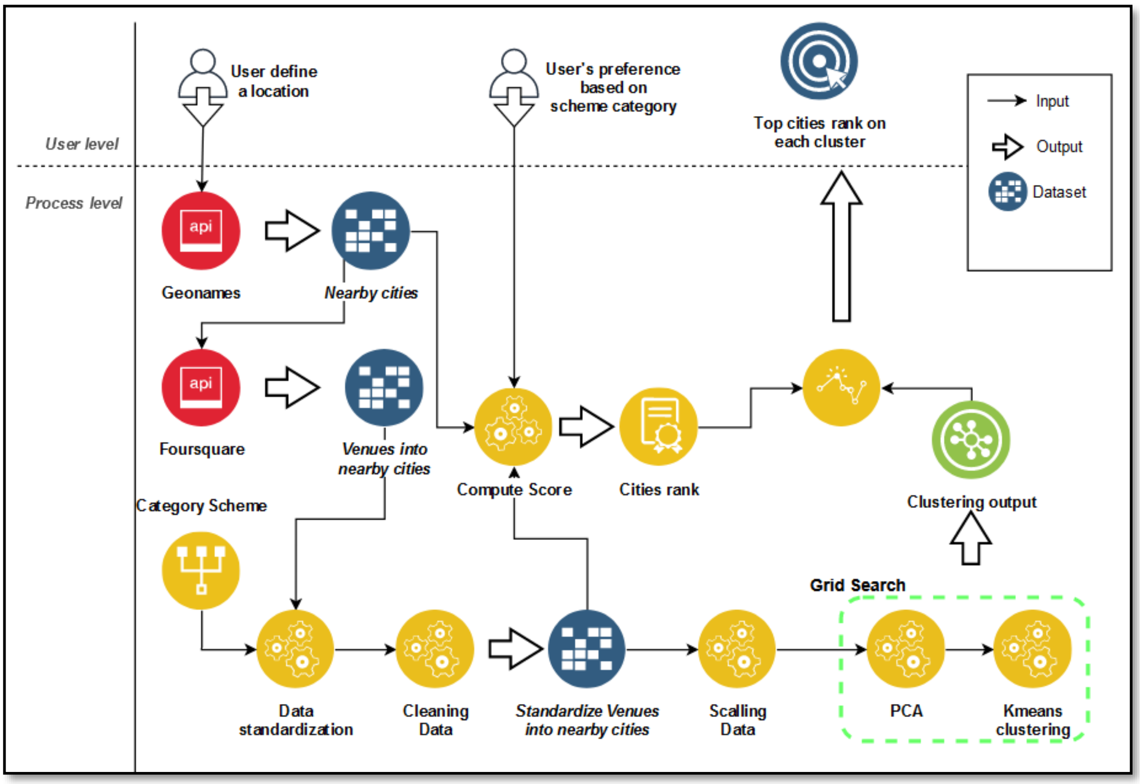


Figure 3: Workflow

Each following sections present step performed in this workflow.

Which Data? Acquisition, Cleaning and explore

DATA ACQUISITION : FULL API FOR MORE FLEXIBILITY

All the data used to give a solution to this problematic are provided by some web service via their API. Mainly three API are used:

- Geonames
- Mapquest
- Foursquare

Each of them and corresponding data mapping are described below:

Mapquest

Mapquest is a platform, which provided geolocation services. It is specialized in Road map and drive problems and have many data on USA. It is primarily based on Geonames web services.

We used principally to retrieve the geo-coordinates of our primary location, for precision. Moreover, as we set this location in France, MapQuest can't provide any data because it has only data on big world city or in USA.

Geonames

The Geonames geographical database is a free database, which contains over 25 million geographical names and consists of over 11 million unique features whereof 4.8 million populated places and 13 million alternate names. Geonames is integrating geographical data such as names of places in various languages, elevation, population and others from various sources.

We used to retrieve all location nearby our primary location of interest. From this data source, we can extract interesting fields, with the *Nearby* function. We are interest with:

- City name
- Latitude
- Longitude
- Population
- Country
- Distance from the searching point.

Foursquare

Foursquare is a provider of geospatial data. It is a user-based service, which collect data from user. With it, we can retrieve from a location all venues in the nearby of it, but also the mark attribute by user and some interesting information on the venue, like things proposed, photo, users information,…

We used it by the *explore* API to retrieve all venues in a radius around a city. Specially we used:

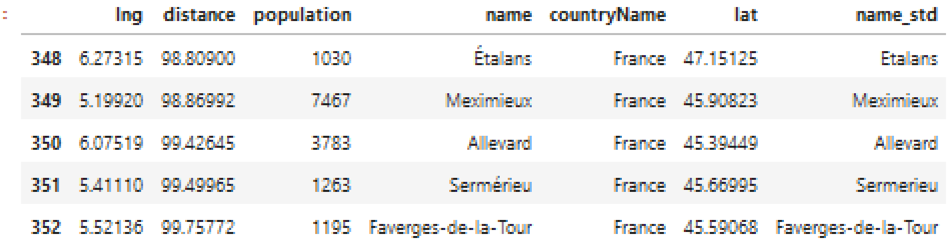
- City name
- Latitude of the city
- Longitude of the city
- Venue
- Venue category
- Venue Id
- Venue latitude
- Venue longitude

We also use it to retrieve the mark of some venues.

Dataset

With these three data source, we can build two principal datasets:

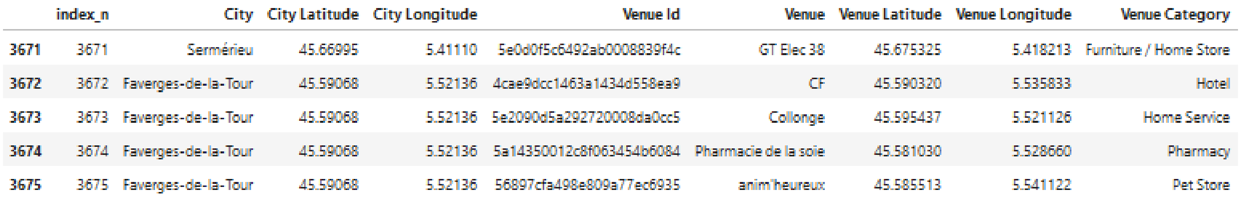
1. First contains all cities around our primary location.



	lng	distance	population	name	countryName	lat	name_std
348	6.27315	98.80900	1030	Étalans	France	47.15125	Etalans
349	5.19920	98.86992	7467	Meximieux	France	45.90823	Meximieux
350	6.07519	99.42645	3783	Allevard	France	45.39449	Allevard
351	5.41110	99.49965	1263	Serméríeu	France	45.66995	Sermerieu
352	5.52136	99.75772	1195	Faverges-de-la-Tour	France	45.59068	Faverges-de-la-Tour

Table 1: Cities dataset

The second contains all different venues available with Foursquare for each city.



	index_n	City	City Latitude	City Longitude	Venue Id	Venue	Venue Latitude	Venue Longitude	Venue Category
3671	3671	Serméríeu	45.66995	5.41110	5e0d0f5c6492ab0008839f4c	GT Elec 38	45.675325	5.418213	Furniture / Home Store
3672	3672	Faverges-de-la-Tour	45.59068	5.52136	4cae9dcc1463a1434d558ea9	CF	45.590320	5.535833	Hotel
3673	3673	Faverges-de-la-Tour	45.59068	5.52136	5e2090d5a292720008da0cc5	Collonge	45.595437	5.521126	Home Service
3674	3674	Faverges-de-la-Tour	45.59068	5.52136	5a14350012c8f063454b6084	Pharmacie de la soie	45.581030	5.528660	Pharmacy
3675	3675	Faverges-de-la-Tour	45.59068	5.52136	56897cfa498e809a77ec6935	anim'heureux	45.585513	5.541122	Pet Store

Table 2: Venues dataset

With our primary setting, we could retrieve 353 cities, and 3694 different venues.

Cleaning and prepare the data

Removing unwanted venue category

We remove from the venues' dataset some venue whose are not useful or too imprecise. For example, we do not want to use *Toll Booth* or *Border,* as they give no information. Moreover, some venues have an imprecise category, like '*Multiplex*'.

Exactly we remove from our sample:

- Border Crossing
- Toll Booth
- Tunnel
- Trade School
- Multiplex
- Rest Area
- Factory
- Construction & Landscaping
- Auto Dealership
- Neighborhood
- Toll Plaza

Scheme standardization

Then we apply a scheme standardization to Venue Category.

Scheme standardization is a technics that consist to assign to a specific value another value, either to correct it or to regroup some value.

We build a category scheme based on all distinct category in our sample (255).

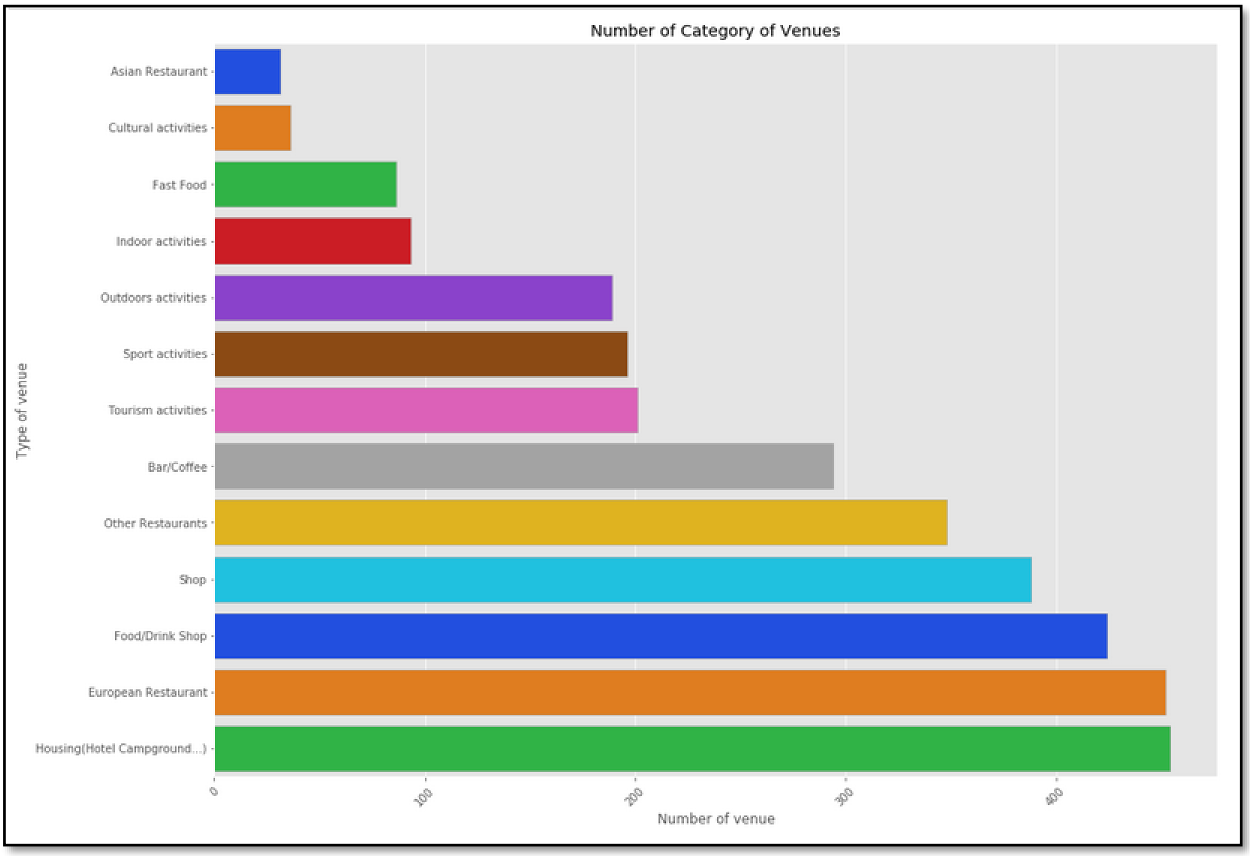
Our scheme allow regrouping value into 14 distinct category:

- Bar/Coffee
- Cultural activities
- Sport activities: *all sports activities, indoors and outdoors.*
- Indoors activities: *all activities whose take place in a closed place (like Casino, arcade game, …) and not considered like sport*
- Outdoors activities: *all activities whose take place outdoors (like Garden, trail, beach, mountain…) and not considered like sport*
- Tourism activities: *all activities commonly reference as tourism, like historic monument, bay view, …)*
- Commodities: *all place and venues like lawyer, town hall, police station, train station, …)*
- Food/Drink Shop: *shop which provide food, drink or both*
- Shop: *all miscellaneous shops (beauty, car, flower, …)*
- Housing (Hotel, Campground…)
- Fast Food
- Asian Restaurant[1]
- European Restaurant
- Other Restaurant

This scheme is arbitrary and not contains all venue category available in Foursquare (approximately 953).

Moreover, for our analysis, we consider that the category '*Commodities'* is not usefull in the decision process of someone for his vacancy. So we remove all venues in this category.

After applying our scheme on the dataset, we reduce the number of category from 255 to 13:

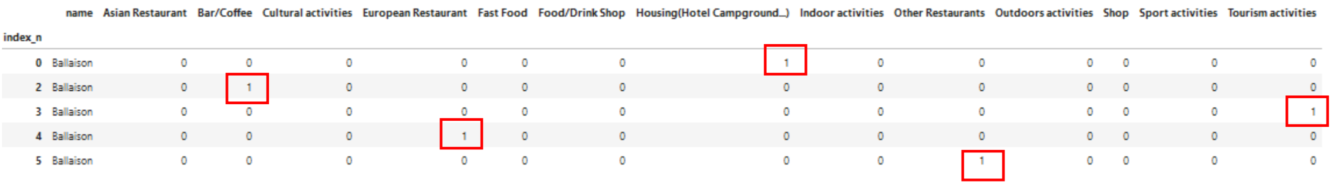


[1] Restaurant a regroup by their geographical origin.

One Hot encoding from standardize venues' category

We apply the One hot encoding method on the Venues dataset.

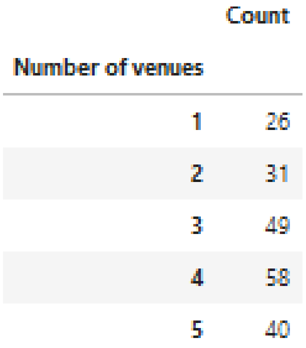
For each venue, we create dummy variable. We create as many dummy variable as category of venue, with 1 if the venue is in the category, else 0.



index_n	name	Asian Restaurant	Bar/Coffee	Cultural activities	European Restaurant	Fast Food	Food/Drink Shop	Housing(Hotel Campground...)	Indoor activities	Other Restaurants	Outdoors activities	Shop	Sport activities	Tourism activities
0	Ballaison	0	0	0	0	0	0	1	0	0	0	0	0	0
2	Ballaison	0	1	0	0	0	0	0	0	0	0	0	0	0
3	Ballaison	0	0	0	0	0	0	0	0	0	0	0	0	1
4	Ballaison	0	0	0	1	0	0	0	0	0	0	0	0	0
5	Ballaison	0	0	0	0	0	0	0	0	1	0	0	0	0

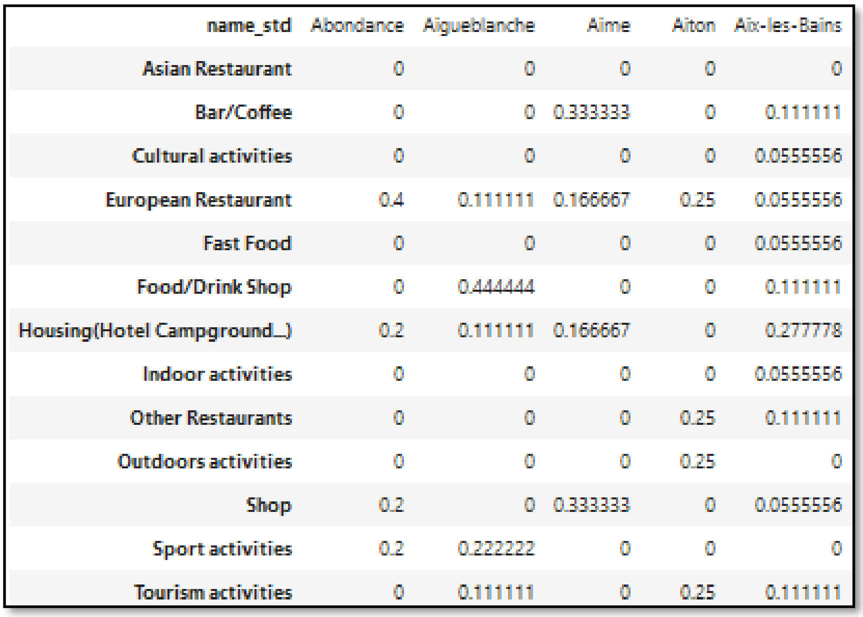
For example, in the city called *Ballaison*, we have some venues (each row is a venue) and the dummy (equal 1) identifies the category of the venue.

Then we before apply the one hot encoding that consist of regroup all venues by city and calculate a mean on the city, we remove city with only one venue. If we don't remove it, the mean calculate in the one hot is 1 and cause noise in the next step of our analyse.



	Count
Number of venues	
1	26
2	31
3	49
4	58
5	40

Finally, with OneHot encoding, we obtain a set of data like below *(transpose version for a better readability)*:



name_std	Abondance	Aigueblanche	Aime	Aiton	Aix-les-Bains
Asian Restaurant	0	0	0	0	0
Bar/Coffee	0	0	0.333333	0	0.111111
Cultural activities	0	0	0	0	0.0555556
European Restaurant	0.4	0.111111	0.166667	0.25	0.0555556
Fast Food	0	0	0	0	0.0555556
Food/Drink Shop	0	0.444444	0	0	0.111111
Housing(Hotel Campground...)	0.2	0.111111	0.166667	0	0.277778
Indoor activities	0	0	0	0	0.0555556
Other Restaurants	0	0	0	0.25	0.111111
Outdoors activities	0	0	0	0.25	0
Shop	0.2	0	0.333333	0	0.0555556
Sport activities	0.2	0.222222	0	0	0
Tourism activities	0	0.111111	0	0.25	0.111111

Figure 4: Output of Onehot encoding

For example, for the city called '*Aigueblanche*', 44.44% of its venues are in '*Food/Drink Shop*' category of venue.

After this step, we are almost ready to apply our methodology of clustering.

Data standardization: Scaling

In next step, we will used Kmeans as clustering algorithm to divide out data and group them. We had used an arbitrary standardization, which had altered data structure of our sample.

To fix this issue, we will use Principal Component Analysis in prior of the kmeans algorithm, in order to reduce the feature space and de-correlate data. However, PCA is sensitive to inputs, and to have a good feature space reduction, we will scale our data previously created by OneHot encoding methods.

Scaling data consist for a variable to subtract it a measure of central tendency, like the mean, and then divide it by a measure of variability, like the standard deviation.

As it exist multiple scaling method, and each of them has different pro and cons and results depends of the sample of data, we try some of them. We compare results with PCA and k-means and how scaling change the nature of data. Below, we described all scaling method tested.

Z-score

Basic and most known scaling methods is the Z-score:

$$x_i' = \frac{x_i - \bar{x}}{\sigma_x}$$

Where $\bar{x}$ is the sample of the mean for the variable$x$, and σ_x is the standard deviation of the sample for the variable x.

This scaling method is sensitive to outliers as it used means and standard deviation. In consequence, it is not robust.

Median / MAD

Median/MAD (Median Absolute Deviation) is more robust. It used the median as measure of central tendency and the MAD as measure of variability.

$$x_i^{MAD} = \frac{x_i - median(x)}{MAD(x)}$$

With $MAD(x) = median(\,|\,x_i - median(x)\,|\,)$

This method provide a robust scaling. The only point of inconvenient is that it does not retain the input distribution and can create changes in the correlation between data. Below an example of the correlations' increasing in our full dataset:

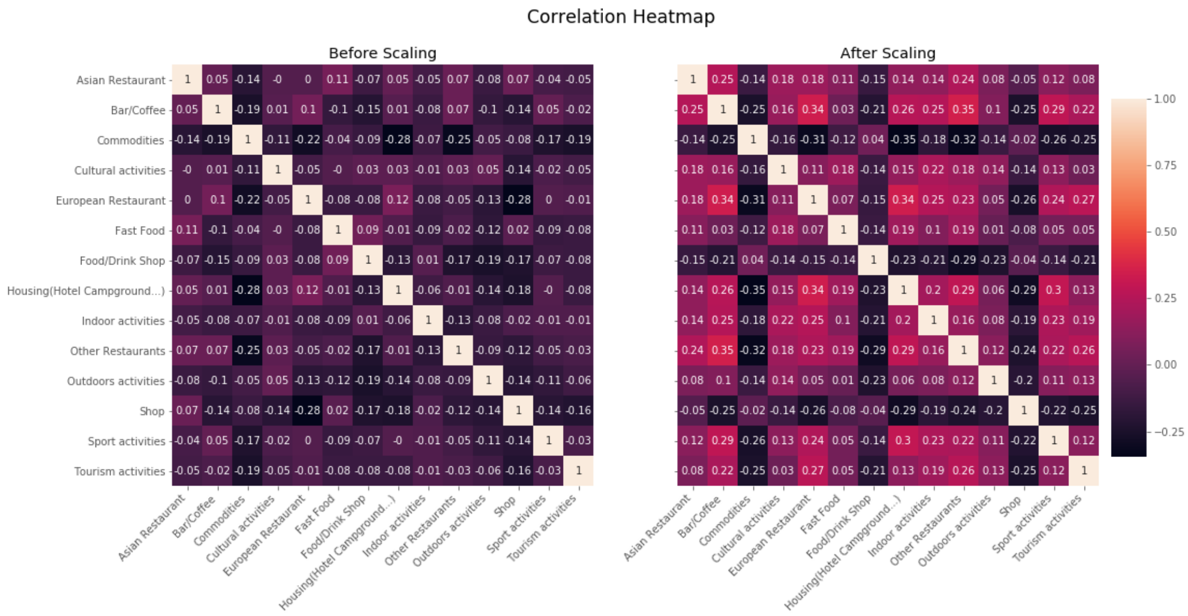


Figure 5: MAD heatmap

Mean/AAD

Mean/AAD is similar to previous, but used mean for central tendancy and average of the absolute deviation:

$$x_i^{AAD} = \frac{x_i - \bar{x}}{AAD(x)}$$

With $AAD(x) = \frac{1}{n}\sum(x_i - \bar{x})$

This method[2] is little less sensitive to outliers than Z-score as we do not square the distance of each data point to its mean (as it is done by the Variance).

We will try this scaling method in the grid search methodology. We will retain the scaling method, which provide the best results for PCA and kmeans. (see next section).

[2] The AAD method in pandas python library is called MAD by language abuse (Mean Absolute deviation), but correspond to the AAD and not MAD.

Methodology

To cluster our sample of cities we will use Kmeans algorithm. As we said previously, this algorithm is little sensitive to inputs, so we decide to reduce the input with Principal Component Analysis as first step.

PRINCIPAL COMPONENT ANALYSIS.

Principal Component Analysis (PCA) is a method used to reduce the dimension of a dataset. For a dataset in *N* dimension, PCA consist to project this high-dimensional data space in a lower dimensional picture, and keep the maximum of information from the data and without redundancy.

For example, a dataset with 50 features can be described with a fewer number of variable, all uncorrelated and with the minimum of information loss.

To avoid weighting down the reading of this document, we do not develop in more details the PCA mathematical concept.

KMEANS CLUSTERING

Kmeans clustering is a clustering algorithm based on a distance metric between clustering center point which is called centroid of the cluster.Algorithm works as follow:

For divide the datapoints in 2 cluster, set two random point as centroid (one for each cluster).

1. Compute the distance metric of all points with the 2 centroids.
2. Assign each point to a cluster, the nearest.
3. Re-compute the center of cluster, which correspond to the mean of all point in the cluster.

Repeat these three step until there is no move in cluster assignment, no move in centroid.

Graphically, this algorithm can be illustrated as below:

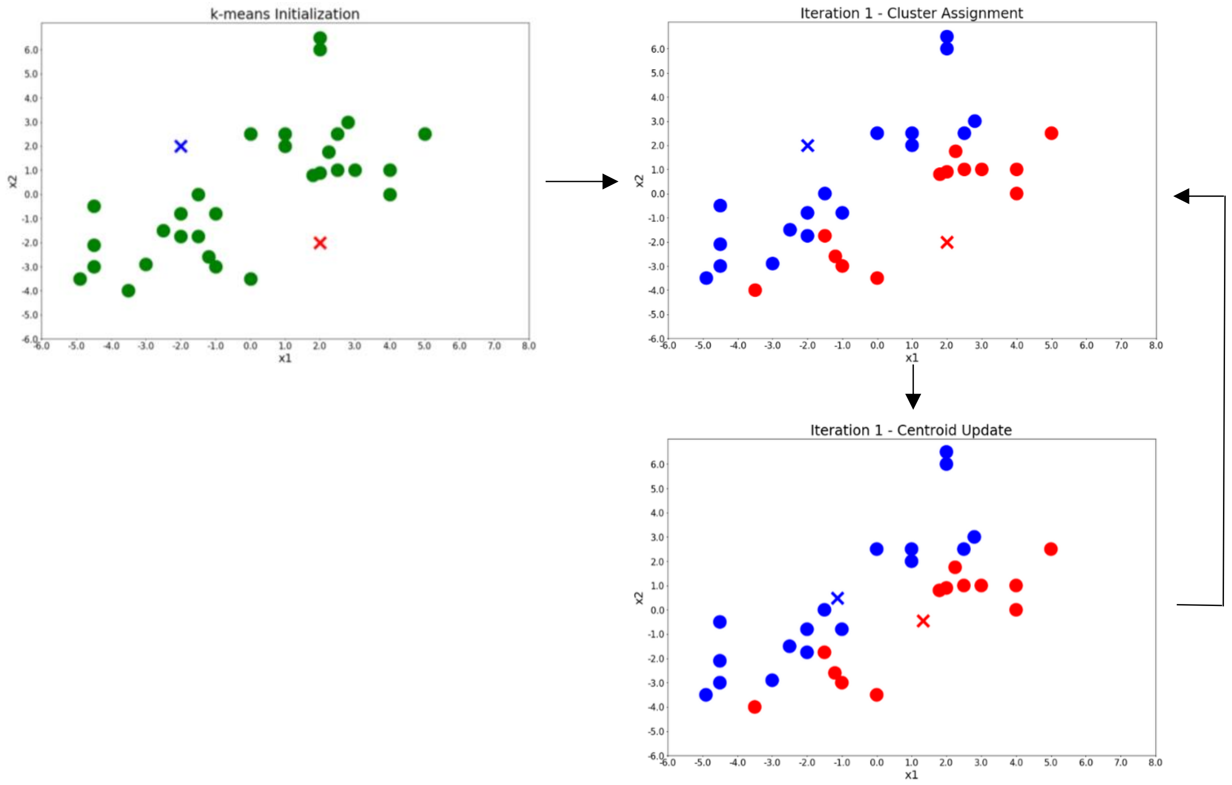


Figure 6: Kmeans, algorithm principles

For the metric used in the distance's computation, we will used the basic Euclidian distance, compute as follow:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Where x and y are into a n- dimension space: $x = (x_1, x_2, \dots x_n)$, $y = (y_1, y_2, \dots y_n)$

Remarks: *it can be possible to use other distance metrics like Manhattan distance, Minkowski, or correlation distance according to the domain.*

We choose kmeans for clustering for its efficiency in computational time. However, kmeans need a number of cluster for its initialization (*k*). This parameter need to be found to have the best clustering of our sample.

ASSESSMENT OF CLUSTERING

To asses that our cluster are good and not overlap, but also to choose the best number of cluster, we will use two assessment metrics in conjunction:

- The average silhouette coefficient
- The Davies-Bouldin indice

Average Silhouette score

The silhouette coefficient is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). It is computed for each datapoint of the dataset as follow:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i,j) \quad and\ b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$$

With k the number of clusters, $d(i,j)$ a distance metric between data points i and j , and C a cluster.

This coefficient is bound between -1 and 1, closest to 1 means a good clustering.

In our case, what interest us is not the silhouette of each data point, but the average of their coefficient:

$$S = \frac{1}{N} \sum_{i=1}^{N} s(i)$$

Davies-Bouldin Index

The Davies-Bouldin indice is an internal evaluation scheme, where the validation of how well the clustering has been done is made using quantities and features inherent to the dataset. It is computed as follow:

For N clusters,

$$DB = \frac{1}{N} \sum_{i=1}^{N} D_i$$

With $D_i = \max_{j \neq i} R_{ij}$, a measure of how good the clustering scheme is.

This value is computed with the measure of separation between each clusters ($M_{i,j}$ which need to be as large as possible for better clustering) and measure of concentration in the cluster i (S_i which need to be as small as possible for better clustering):

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}}$$

$$S_i = \left(\frac{1}{T_i} \sum_{j=1}^{T_i} | X_i - A_i |^p \right)^{\frac{1}{p}} \quad and \quad M_{i,j} = \left\| A_i - A_j \right\|_p = \left(\sum_{k=1}^{n} \left| a_{k,i} - a_{k,} \right|^p \right)^{\frac{1}{p}}$$

With A_i, the centroid of the cluster i, T_i the size of the cluster, p the order for the distance metrics (2 give Euclidian distance), and X_i an n-dimensional feature vectors.

The range of values of this index is $[0 ; +\infty]$. A good clustering implies a value near of 0, and a bad data partitioning is a large value.

We will used this two assessment in our methodology to find the best value of parameters and create our cluster.

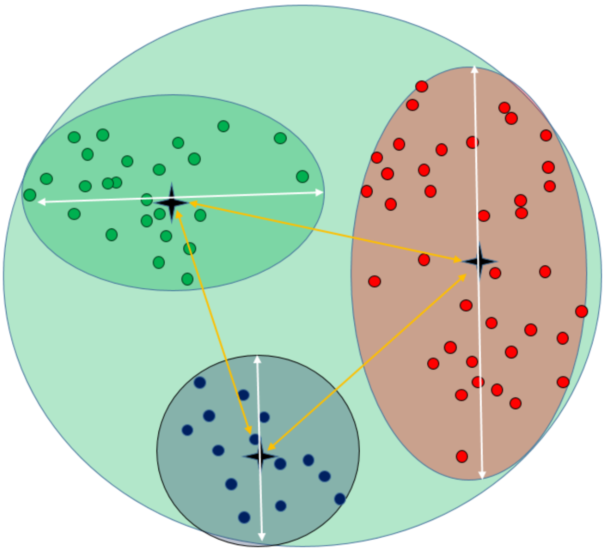
Variance Between/Variance Within

Moreover, we will use the common measure of variance for the clustering, the Between Variance and the Within Variance (also used in ANOVA

For a cloud of point, we can define the below relation on his Variance and its clusters:

$$Total\ Variance = Variance\ Between + Variance\ Within$$

The clustering of the data point don't affect the Variance of the cloud. It is always equal to the sum of Variance Between clusters and the sum of variance within each cluster.



And we can compute a coefficient of R2:

$$R^2 = \frac{Between\ Variance}{Total\ Variance}$$

Closest to 1 indicate a good partitioning of the data.

GRID SEARCH

As we used in conjunction PCA and kmeans, we need to find for our sample which number of features describe the best our sample, and then the best number of cluster. For this we use a small grid search method.

For a couple of value (f, k), number of feature f in PCA and number of cluster k in kmeans, we compute the average silhouette coefficient and the Davies-Bouldin score. With this, we have for a specific value of f a range of value, one for each k tested, and can plot a boxplot of the result for the two metrics. We can focus in a specific value of f for our PCA and then focus on the best value of k for our kmeans algorithm.

We will test a grid with $f \in [1; 5]$ and $k \in [2; 7]$, which lead to compute 30 times the kmeans algorithm for a specific dataset.

Moreover, we perform this exercise of grid search for each scaling method spreviously described. Finaly, this lead to compute 120 times the kmeans algorithm.

CUSTOM SCORING

To propose a small list of city in each cluster, we will use a user scoring-rank. This method is similar to build a score-grid in the event prediction, like in credit scoring. The goal is to have a grid with each modalities used in a model and number of point associated to this modalities. This grid is used to better understand the process of decision making by the model.

To understand this, take the logistic model below :

$$Logit(Y) = c + \beta_0 X_0 + \beta_1 X_1 + \beta_2 X_2 + \gamma_0 Z_0 + \gamma_1 Z_1$$

Where *Logit(Y)* is the ratio of probability, X is a variable discretize in 3 modalities (3 dummy variables) and Z another variable discretize in 2 modalities. Dummy variable are such that only one of them equal 1 for a specific individual, other are equal to 0. $\beta, \gamma\ and\ c$ are our coefficient estimated.

To make a prediction understandable in term of decision process, we can't keep the value estimated. So we transform it in a grid score (on a scale define, like 100) by different simple mathematic operations. Results is something comparable as table display below:

Variable	Modalities	Score
C (intercept)	c	0
X	X_0	20
	X_1	70.5
	X_2	50
Z	Z_0	10.5
	Z_1	25

Hence, to make a prediction for someone, we compute his score by summing the score associated to each of his characteristics. For example if an individual has $X_0\ and\ Z_1$, his score is 0 + 20 + 25 = 45. Following this, it is possible to rank people based on their score.

Then only condition necessary to transform estimation in grid is to have modalities. This does not work with continuous variable (like salary or age). It is necessary to regroup in class.

Remarks: *It's impossible for us to detail the methodology used to build a score grid, this methodology is specific and it's like a secret recipe that we studied and promised to not share.*

Alternative method is use the logistic function to compute the probability (which is similar to the score):

$$\Pr(Y = 1|W) = \frac{1}{1 + e^{-\theta W}}$$

Where $\theta W = c + \beta_0 X_0 + \beta_1 X_1 + \beta_2 X_2 + \gamma_0 Z_0 + \gamma_1 Z_1$.

This other methods give the same results and orders, expressed in probability of choice (∈ [0;1])

In our case, it little bit different. We did not fit any model in our analysis. However, we used scheme to regroup our venues. So we can use these as '*modalities'* and as input variables. Coefficients can be provide directly by user, as user's preference. User can set for each category type a value like -5, -50, 20 … according to what it is more important for him in a choice of a city.

In our case, we create a custom grid score with user's preference based on the scheme value. So we have only 1 variable (Venue Category) and it is divided in category (by the scheme).

In our case, the scoring grid for a user with a set of preference (as example) is:

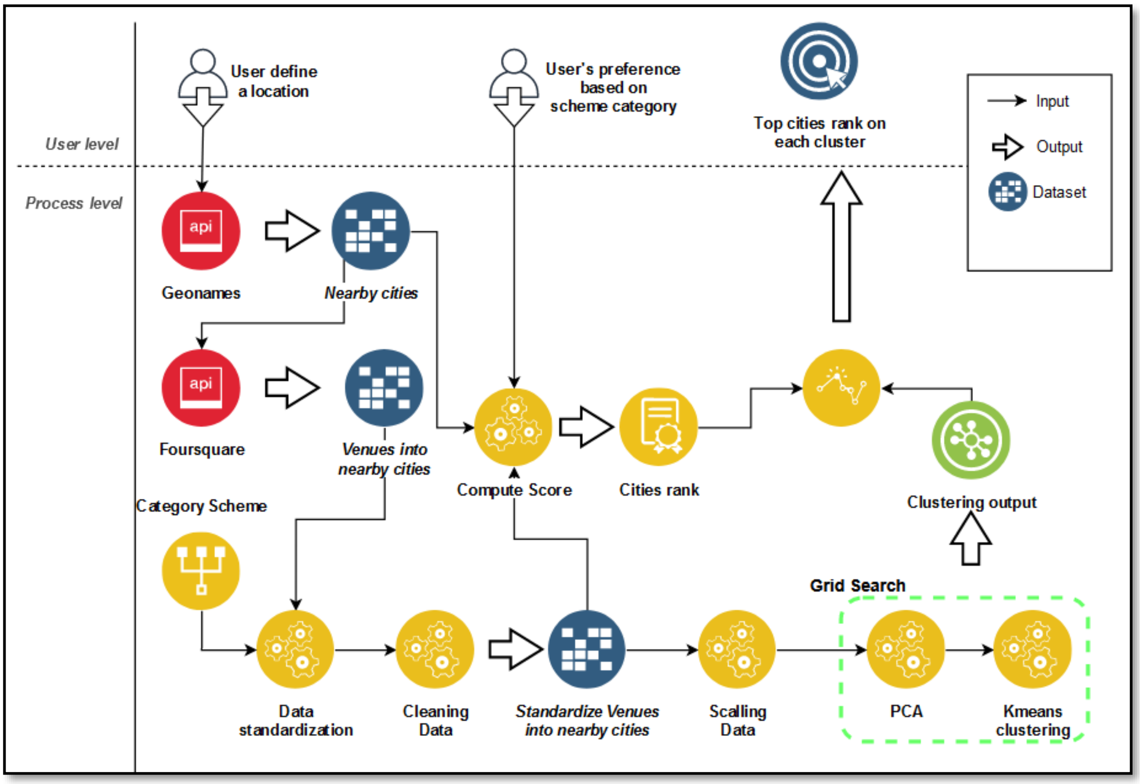
Variable	**Modalities**	**Coefficient** *(= user's preference)*	**Score (scale = 100)**
Venue Category (standardize)	Asian Restaurant	$b_0 = -3$	20.0
	Bar/Coffee	$b_1 = 5$	100.0
	Cultural activities	$b_2 = 4$	90.0
	European Restaurant	$b_3 = 3$	80.0
	FastFood	$b_4 = -5$	0.0
	Food/Drink Shop	$b_5 = 2$	70.0
	Housing (Hotel, Campground...)	$b_6 = 2$	70.0
	Indoor activities	$b_7 = -1$	40.0
	Other Restaurants	$b_8 = -2$	30.0
	Outdoor activities	$b_9 = 4$	90.0
	Shop	$b_{10} = 2$	70.0
	Sports Activities	$b_{11} = 3$	80.0
	Tourism Activities	$b_{12} = 4$	90.0

Then for each city, we compute the score by multiply each value of venue category by the corresponding score. The result is a score on a scale (set to 100 in our case). Then we can use this rank to keep only a small number of city.

Results

SUMMARY

As reminder, to design our solution, we followed the workflow below:



GRID SEARCH RESULTS

Grid search on data not scalled:

Results is not very good. Silhouette score is most of the time inferior to Davies-Bouldin index, implies that that the clustering is bad. We must apply a scalling method to improve results.

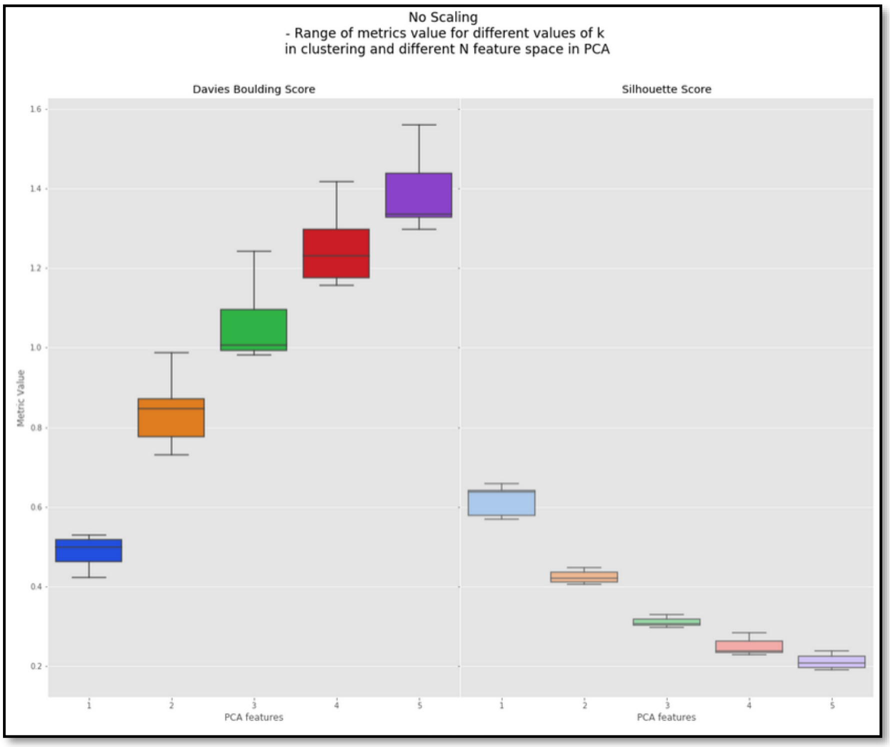
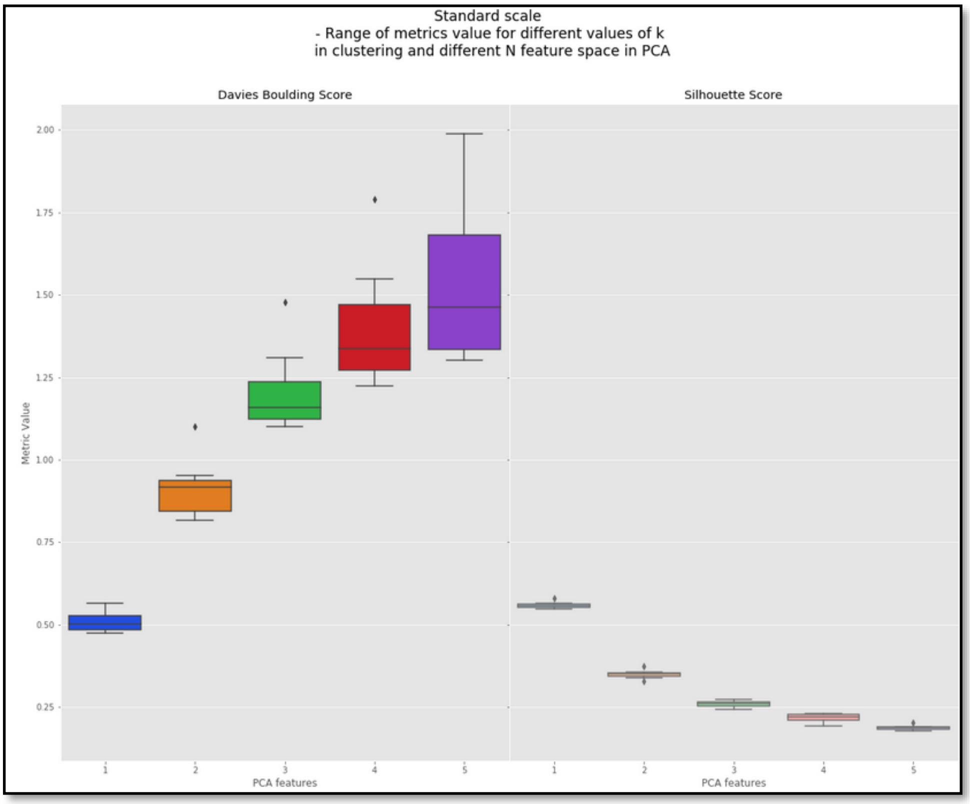


Figure 7: Grid Search on no scaled data

Z-score scaling:



With a standard scaling, results are also not very good. Better than without scaling.

Figure 8: Grid Search on z-score scaled data

Median/MAD scaling:

MAD scalling give better results, but only with a PCA with 1 features.

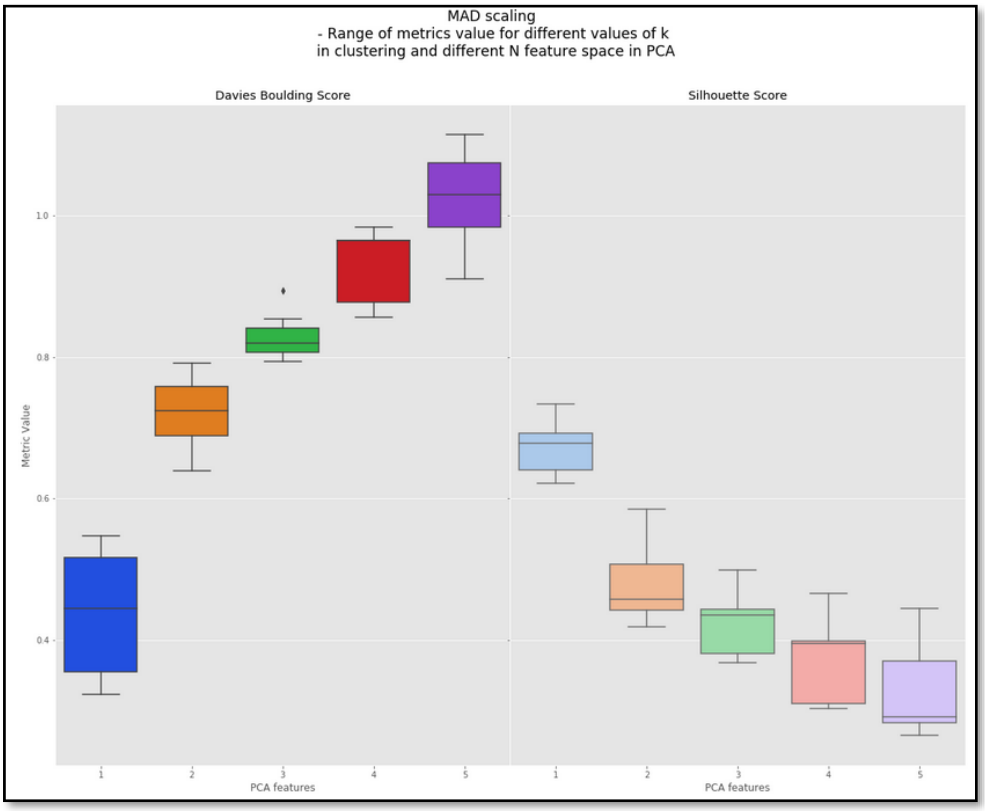


Figure 9: Grid Search on MAD scaled data

Mean/AAD scaling

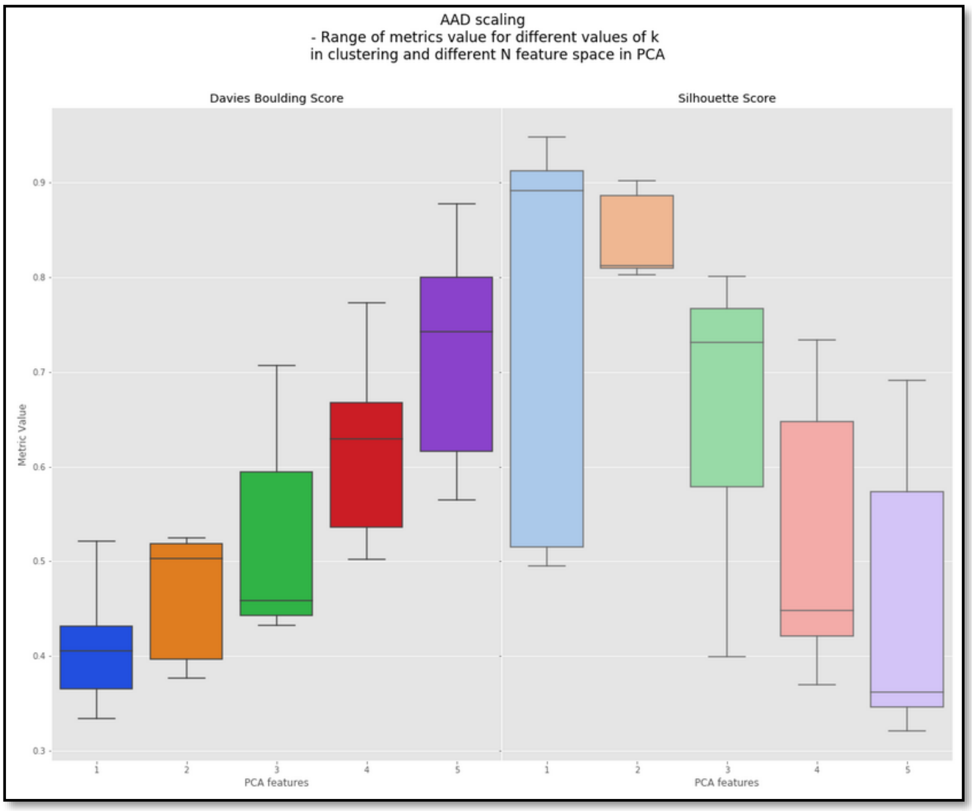


Figure 10: Grid Search on AAD scaled data

It seems that AAD scaling methods give the best results. Silhouette score is superior to Davies-Bouldin index, which is what we want.

We keep it and use 2 feature in PCA in order to have a minimum of interpretability.

Now we find the best number of cluster k:

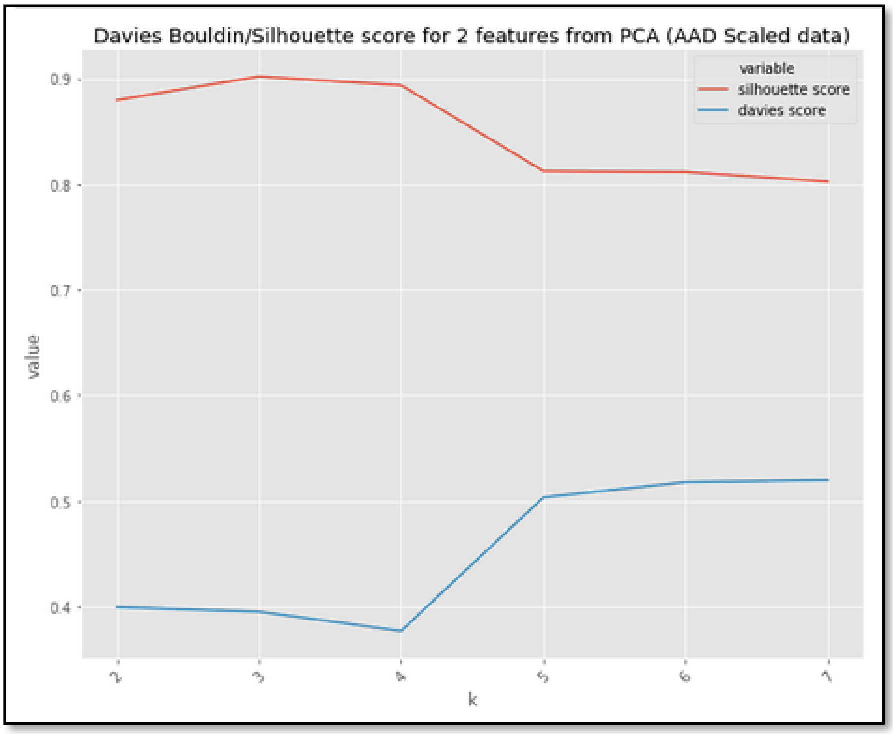


Figure 11: Cluster assesment for AAD sclaed data (PCA 2 features)

We can choose 2, 3 or 4 clusters. One of these give good results. We choose 4 clusters for our kmeans.

Principal Component Analysis / Kmeans

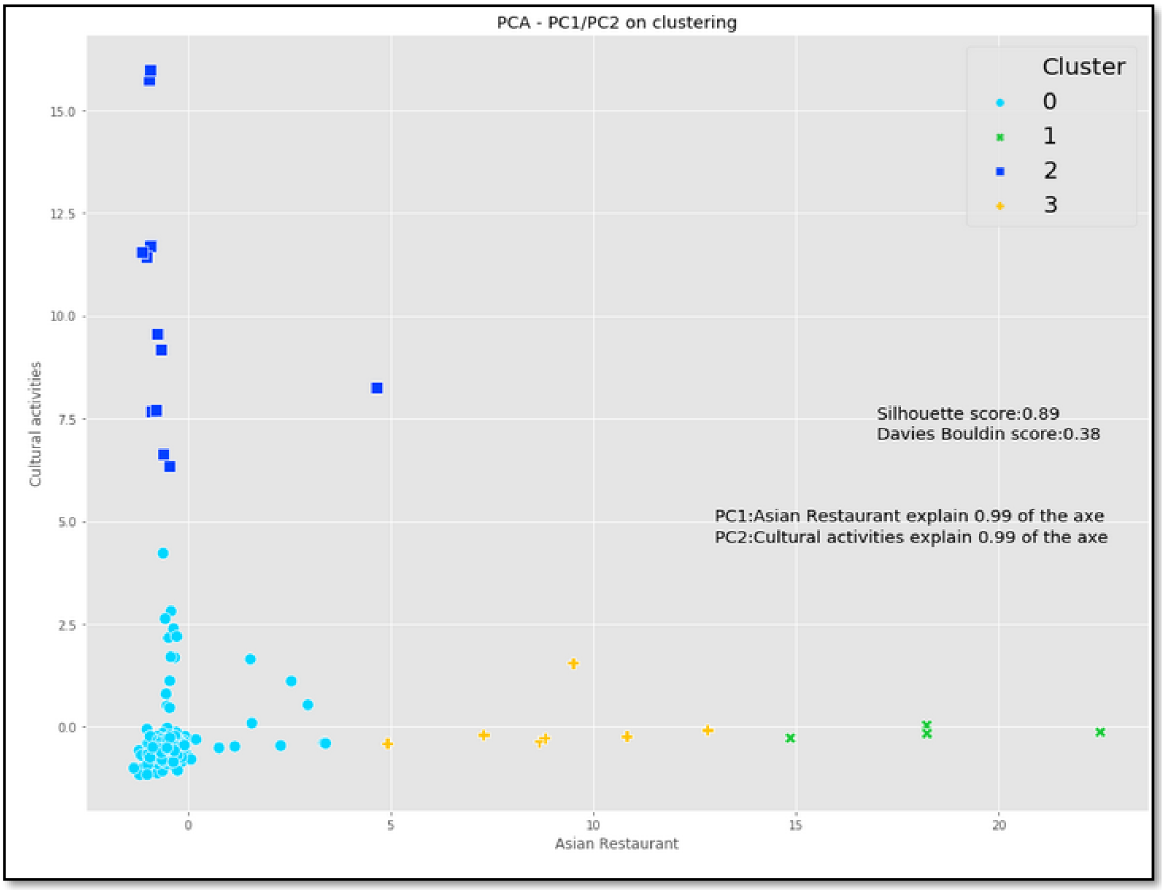


Figure 12: PCA plot - PC1 vs PC2

We can interpret the results of our PCA.

- Cluster 0 contains city with a fewer number of Asian Restaurant and Cultural activities.
- Cluster 1 are cities with lots of Asian restaurant in their venues, and no cultural activities.
- Cluster 2 is the contrary, lots of Cultural activities and not Asian Restaurant.
- Cluster 3 have some Cultural activities and Asian Restaurant.

KMEANS RESULTS

Below the map of our cities. Each color is a cluster.

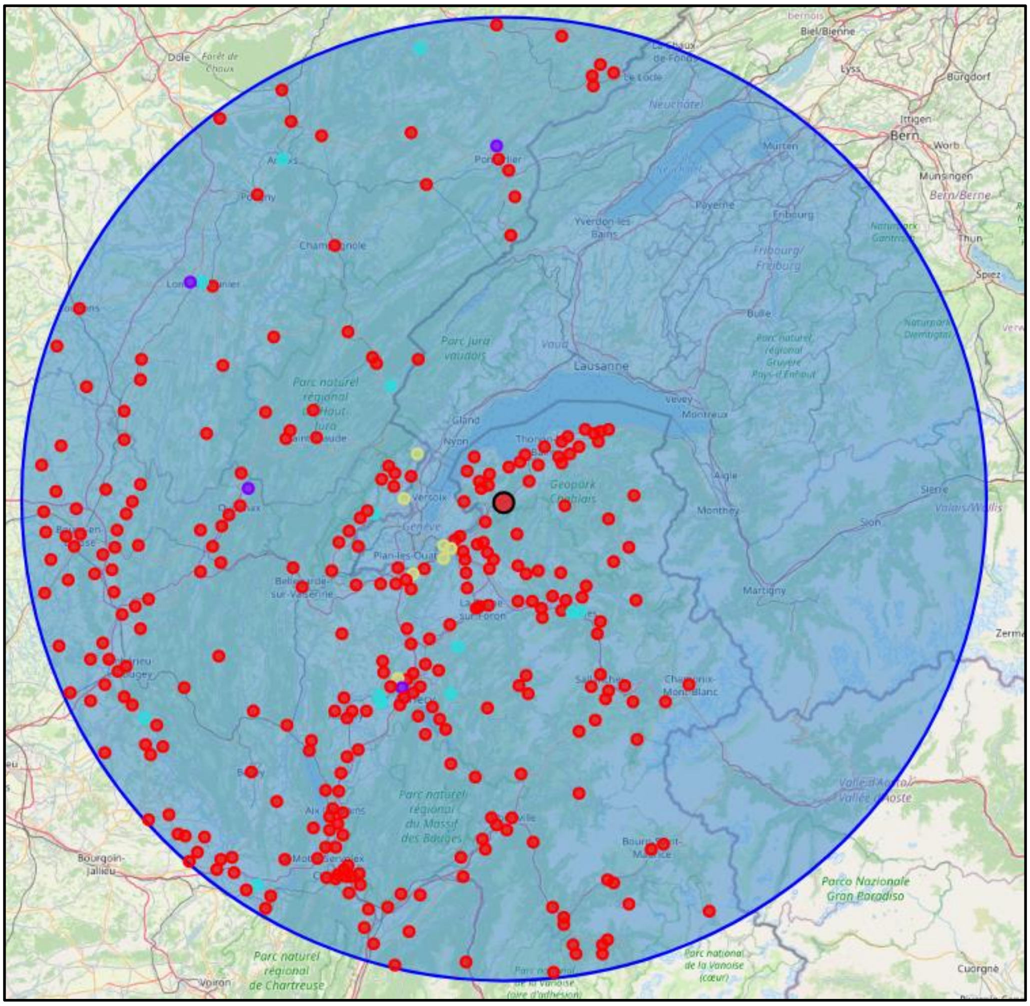
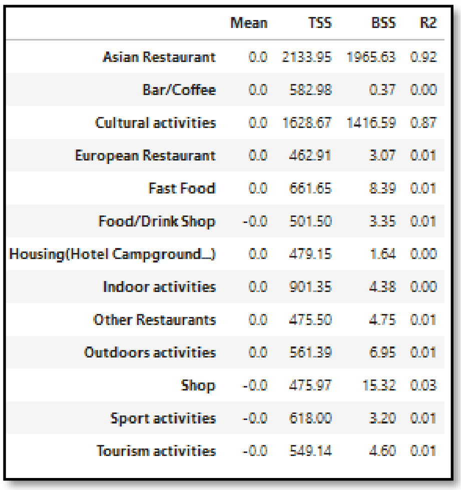


Figure 13: Clustering of cities - Map

As we can see, clustering doesn't retrieve geospatial shape. Cluster data point are sparser and not regroup all in the same place.

If we check the Variance explained by clustering, we retrieved what PCA show us:



	Mean	TSS	BSS	R2
Asian Restaurant	0.0	2133.95	1965.63	0.92
Bar/Coffee	0.0	582.98	0.37	0.00
Cultural activities	0.0	1628.67	1416.59	0.87
European Restaurant	0.0	462.91	3.07	0.01
Fast Food	0.0	661.65	8.39	0.01
Food/Drink Shop	-0.0	501.50	3.35	0.01
Housing(Hotel Campground...)	0.0	479.15	1.64	0.00
Indoor activities	0.0	901.35	4.38	0.00
Other Restaurants	0.0	475.50	4.75	0.01
Outdoors activities	0.0	561.39	6.95	0.01
Shop	-0.0	475.97	15.32	0.03
Sport activities	-0.0	618.00	3.20	0.01
Tourism activities	-0.0	549.14	4.60	0.01

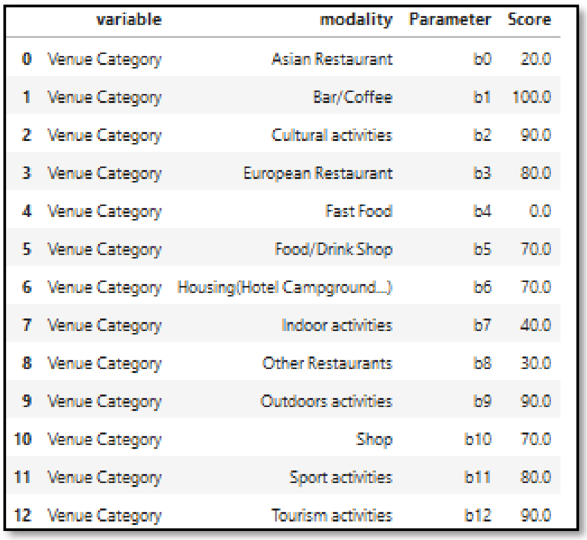
Table 3: Variance explained in clusters

Our Clusters explain well the variance of Asian Restaurant and Cultural activity (92% and 87% respectively).

Scoring grid

Our score grid is compute with this set of preferences and on a scale of 100:

Modalities	Coefficient *(= user's preference)*
Asian Restaurant	$b_0 = -3$
Bar/Coffee	$b_1 = 5$
Cultural activities	$b_2 = 4$
European Restaurant	$b_3 = 3$
FastFood	$b_4 = -5$
Food/Drink Shop	$b_5 = 2$
Housing (Hotel, Campground...)	$b_6 = 2$
Indoor activities	$b_7 = -1$
Other Restaurants	$b_8 = -2$
Outdoor activities	$b_9 = 4$
Shop	$b_{10} = 2$
Sports Activities	$b_{11} = 3$
Tourism Activities	$b_{12} = 4$



	variable	modality	Parameter	Score
0	Venue Category	Asian Restaurant	b0	20.0
1	Venue Category	Bar/Coffee	b1	100.0
2	Venue Category	Cultural activities	b2	90.0
3	Venue Category	European Restaurant	b3	80.0
4	Venue Category	Fast Food	b4	0.0
5	Venue Category	Food/Drink Shop	b5	70.0
6	Venue Category	Housing(Hotel Campground...)	b6	70.0
7	Venue Category	Indoor activities	b7	40.0
8	Venue Category	Other Restaurants	b8	30.0
9	Venue Category	Outdoors activities	b9	90.0
10	Venue Category	Shop	b10	70.0
11	Venue Category	Sport activities	b11	80.0
12	Venue Category	Tourism activities	b12	90.0

Table 4: Score Grid

We can compute for each city of our sample a score with this grid. Distribution of this score over clusters are plotted below:

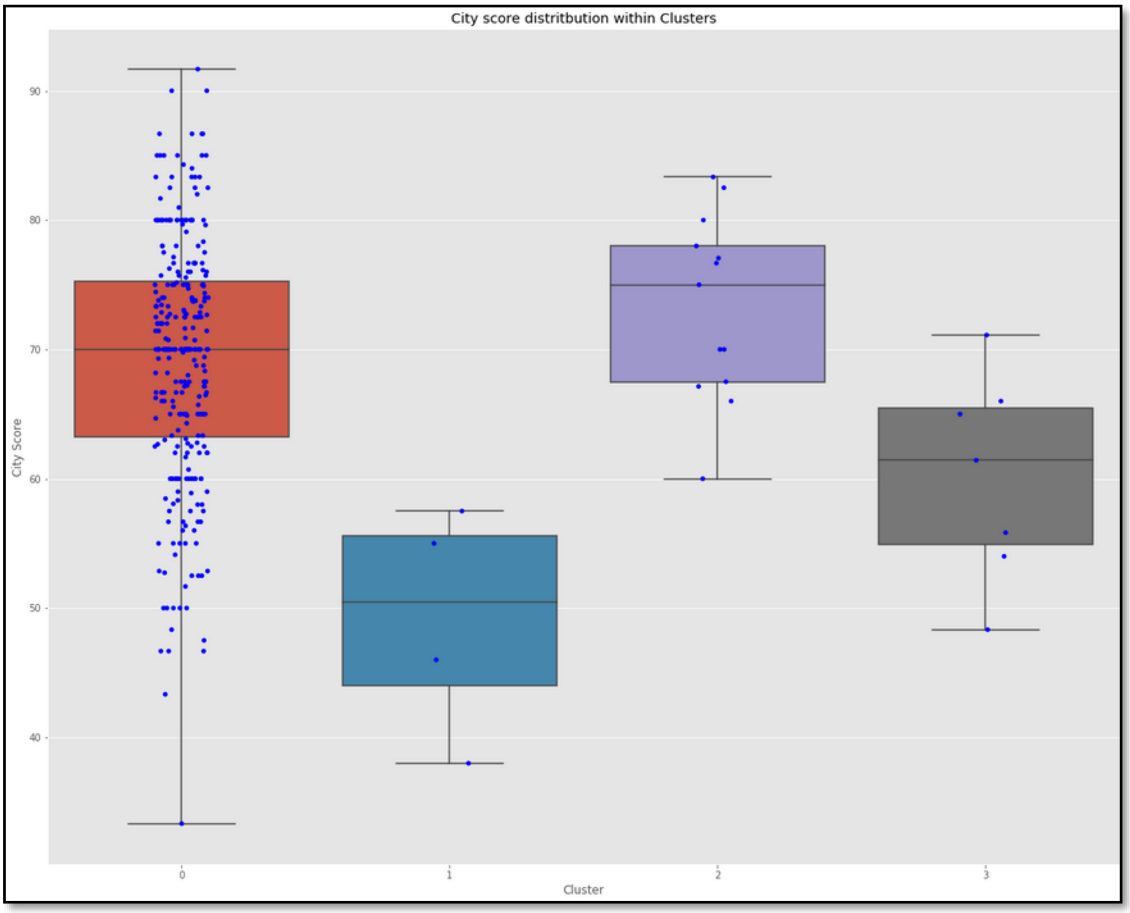
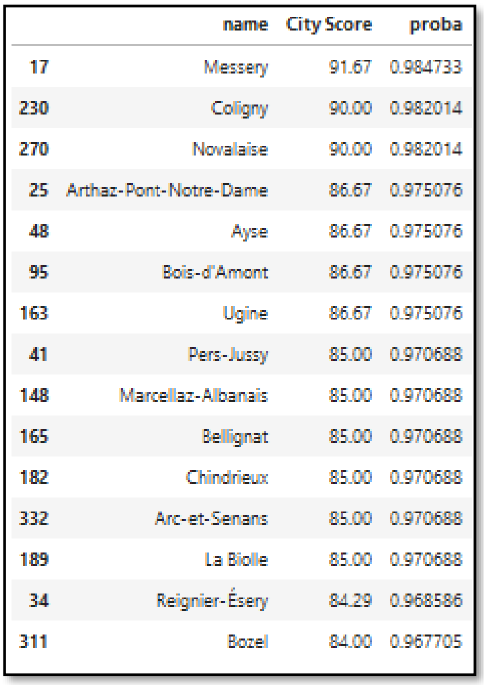


Figure 14: Score distribution through clusters

We also compute the probability associated with logistic function (see previous section). We can see that the score method give same results but on a human scale interpretable.

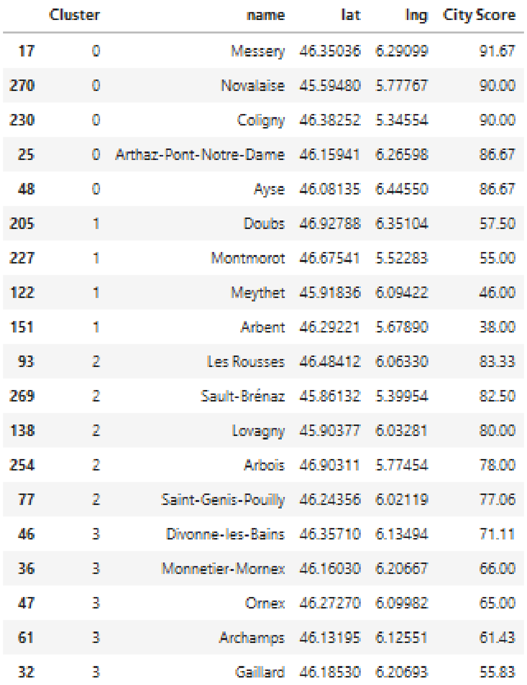


	name	City Score	proba
17	Messery	91.67	0.984733
230	Coligny	90.00	0.982014
270	Novalaise	90.00	0.982014
25	Arthaz-Pont-Notre-Dame	86.67	0.975076
48	Ayse	86.67	0.975076
95	Bois-d'Amont	86.67	0.975076
163	Ugine	86.67	0.975076
41	Pers-Jussy	85.00	0.970688
148	Marcellaz-Albanais	85.00	0.970688
165	Bellignat	85.00	0.970688
182	Chindrieux	85.00	0.970688
332	Arc-et-Senans	85.00	0.970688
189	La Biolle	85.00	0.970688
34	Reignier-Ésery	84.29	0.968586
311	Bozel	84.00	0.967705

Table 5: City Score vs probability of choice

RANK OF CITY

Now we keep a maximum of 5 cities in each clusters, those whose are the highest score:



	Cluster	name	lat	lng	City Score
17	0	Messery	46.35036	6.29099	91.67
270	0	Novalaise	45.59480	5.77767	90.00
230	0	Coligny	46.38252	5.34554	90.00
25	0	Arthaz-Pont-Notre-Dame	46.15941	6.26598	86.67
48	0	Ayse	46.08135	6.44550	86.67
205	1	Doubs	46.92788	6.35104	57.50
227	1	Montmorot	46.67541	5.52283	55.00
122	1	Meythet	45.91836	6.09422	46.00
151	1	Arbent	46.29221	5.67890	38.00
93	2	Les Rousses	46.48412	6.06330	83.33
269	2	Sault-Brénaz	45.86132	5.39954	82.50
138	2	Lovagny	45.90377	6.03281	80.00
254	2	Arbois	46.90311	5.77454	78.00
77	2	Saint-Genis-Pouilly	46.24356	6.02119	77.06
46	3	Divonne-les-Bains	46.35710	6.13494	71.11
36	3	Monnetier-Mornex	46.16030	6.20667	66.00
47	3	Ornex	46.27270	6.09982	65.00
61	3	Archamps	46.13195	6.12551	61.43
32	3	Gaillard	46.18530	6.20693	55.83

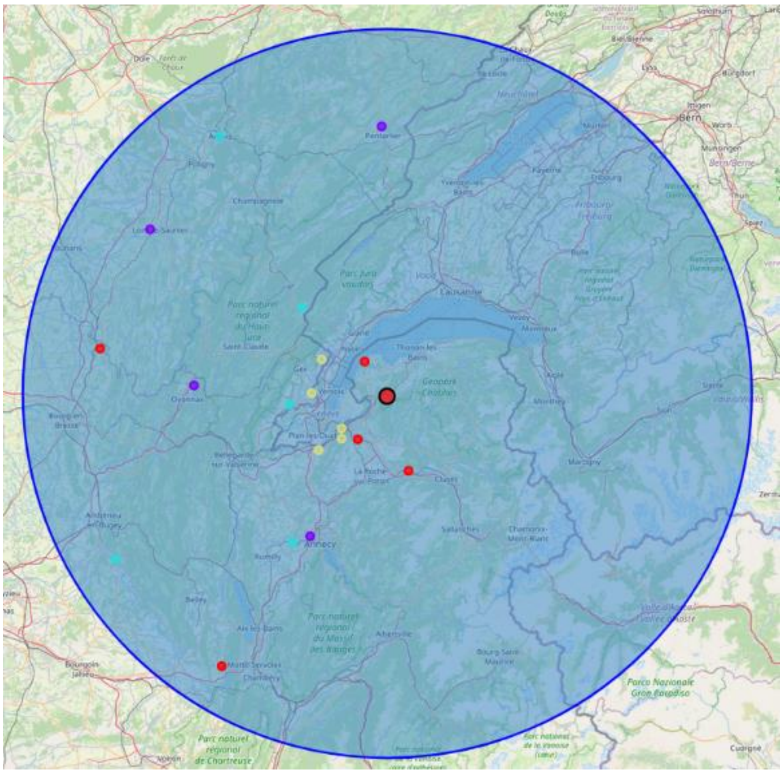
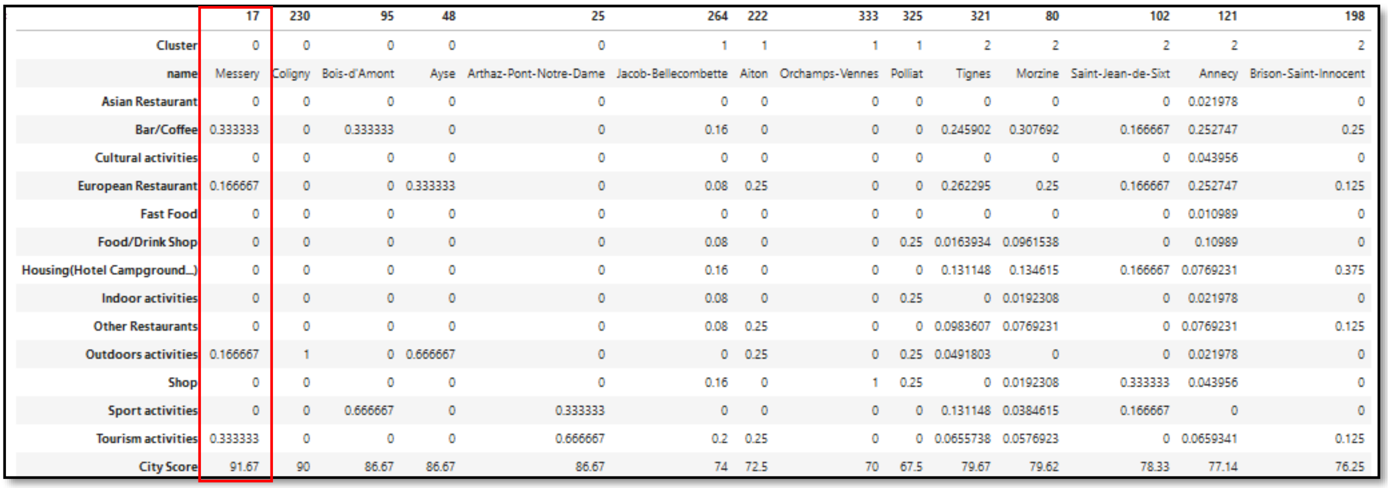


Figure 15: Cities retained



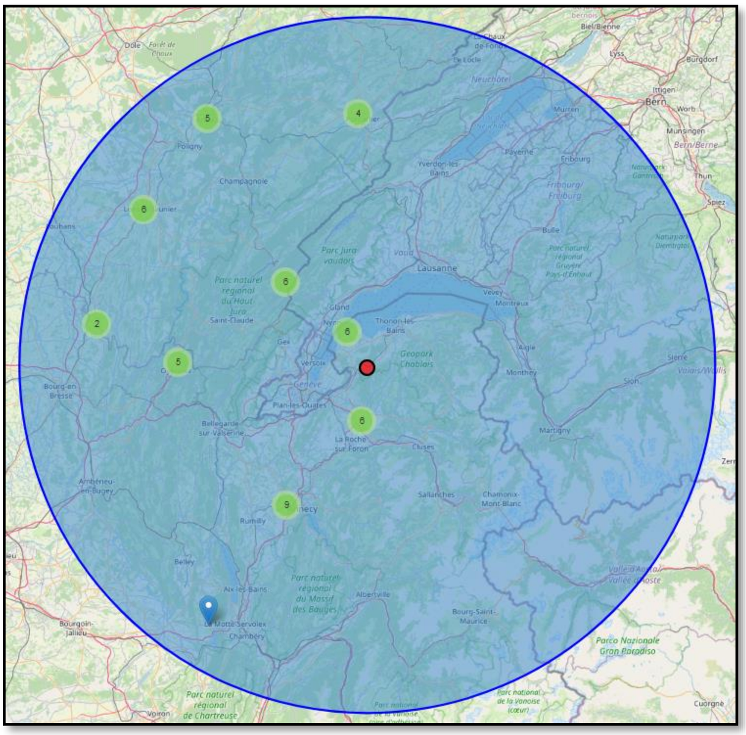
	17	230	95	48	25	264	222	333	325	321	80	102	121	198
Cluster	0	0	0	0	0	1	1	1	1	2	2	2	2	2
name	Messery	Coligny	Bois-d'Amont	Ayse	Arthaz-Pont-Notre-Dame	Jacob-Bellecombette	Aiton	Orchamps-Vennes	Polliat	Tignes	Morzine	Saint-Jean-de-Sixt	Annecy	Brison-Saint-Innocent
Asian Restaurant	0	0	0	0	0	0	0	0	0	0	0	0	0.021978	0
Bar/Coffee	0.333333	0	0.333333	0	0	0.16	0	0	0	0.245902	0.307692	0.166667	0.252747	0.25
Cultural activities	0	0	0	0	0	0	0	0	0	0	0	0	0.043956	0
European Restaurant	0.166667	0	0	0.333333	0	0.08	0.25	0	0	0.262295	0.25	0.166667	0.252747	0.125
Fast Food	0	0	0	0	0	0	0	0	0	0	0	0	0.010989	0
Food/Drink Shop	0	0	0	0	0	0.08	0	0	0.25	0.0163934	0.0961538	0	0.10989	0
Housing(Hotel Campground...)	0	0	0	0	0	0.16	0	0	0	0.131148	0.134615	0.166667	0.0769231	0.375
Indoor activities	0	0	0	0	0	0.08	0	0	0.25	0	0.0192308	0	0.021978	0
Other Restaurants	0	0	0	0	0	0.08	0.25	0	0	0.0983607	0.0769231	0	0.0769231	0.125
Outdoors activities	0.166667	1	0	0.666667	0	0	0.25	0	0.25	0.0491803	0	0	0.021978	0
Shop	0	0	0	0	0	0.16	0	1	0.25	0	0.0192308	0.333333	0.043956	0
Sport activities	0	0	0.666667	0	0.333333	0	0	0	0	0.131148	0.0384615	0.166667	0	0
Tourism activities	0.333333	0	0	0	0.666667	0.2	0.25	0	0	0.0655738	0.0576923	0	0.0659341	0.125
City Score	91.67	90	86.67	86.67	86.67	74	72.5	70	67.5	79.67	79.62	78.33	77.14	76.25

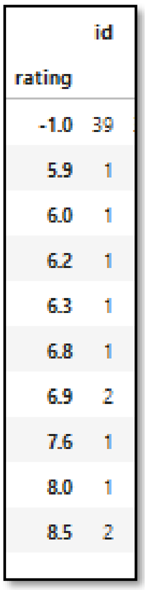
We can easily decompose and interpret the score of each city retained. For example, the city of Messery:

Venue Category	**% of the venues in the city (1)**	**Points (2)**	$(1) \times (2)$
Bar/Coffee	*33%*	*100*	*33*
European Restaurant	*17%*	*80*	*13.6*
Outdoors activities	*17%*	*90*	*15.3*
Tourism activities	*33%*	*90*	*29.7*
TOTAL	100%	-	91.6

For each city in with the highest score, we had retrieve the rating of each venues. For our cities retain, we have 73 venues. But we can retrieve a maximum of 50 rating (limit of the API for our account).

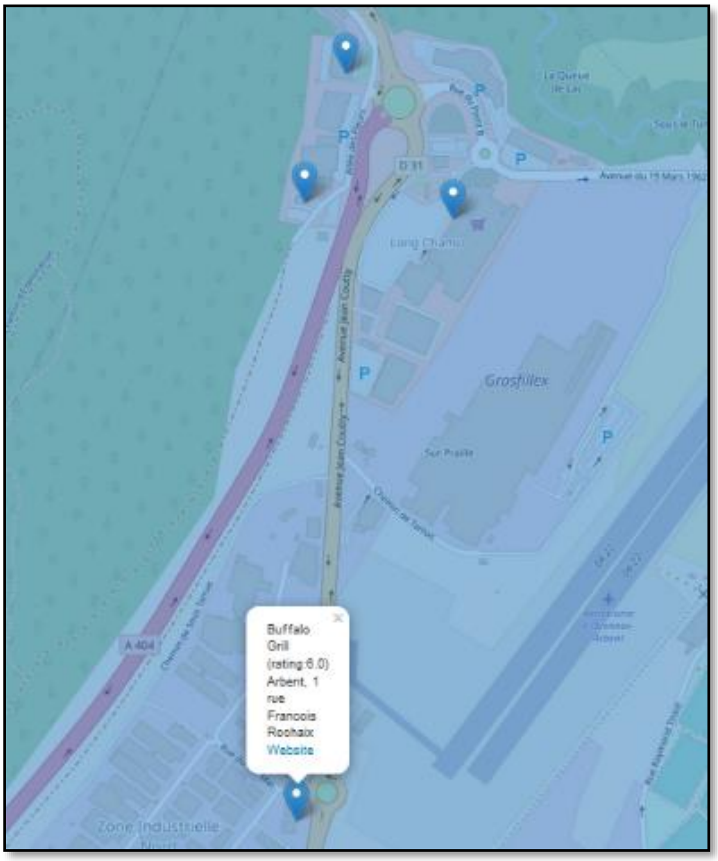
In this 50 venues there are some of them which are not rated yet.





	id
rating	
-1.0	39
5.9	1
6.0	1
6.2	1
6.3	1
6.8	1
6.9	2
7.6	1
8.0	1
8.5	2

Figure 16: Venues' rating



Buffalo Grill (rating:6.0) Arbent, 1 rue Francois Rochaix Website

Tête de l'écutieux 1627 m
Le Môle 1863 m
Petit Môle 1534 m
Ayse

Le Mole (rating:nan) Ayse, Up the hill Website

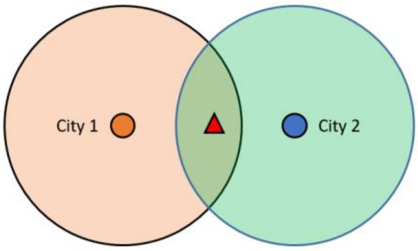
Discussion

Following the results, we can discuss about some point of attention. In our solution, there are some possible leverage to change the results, first is the scheme used. The scheme standardization allow us a dimensionally reduction of the input by grouping them. But this scheme is totally arbitrary and other scheme could be test to compare the performance. A different scheme could lead to an other number of feature in PCA.

Another leverage is the data standardization used. We used Mean/AAD scale in our sample, but other standardization could be implemented, like double-sigmoid function or tanh-estimator. They could lead to another result. Moreover, according to the sample the results could be very different.

Then, in our methodology, we use only kmeans, as it is a faster algorithm. But maybe hierarchic algorithm could be good, or Density based (DBSCAN, BIRCH).

Another point is in the data quality. In the phase of retrieve the venues nearby a location, some of them can be retrieve twice, especially when two cities are close. This lead to an overlap on the radius of the search and it is difficult to find the true location of the venue.



In our case, we cannot remove the duplicate to not overweighting some city, as there is a probability of 1/n to find the true city (n is the number of city whose overlap).

The data quality is also very important in our case. As we used Web service to create our dataset, there is a possibility of the data collect are not up to date according to the provider. Moreover, using a web service like Foursquare API had led to rise another issue, the lack of data. Indeed, in some location like the one we set in the campaign at the beginning of this project have poor in data. The reason is simply cause by the fact that Foursquare use users to collect its data. If in this location, a small number of people use it, we retrieve a small sample of data, not very representative of the location.

Conclusion

As conclusion of our study, we can provide a small list of cities to a user to plan his vacation during the Covid-19 crisis.

We design a small workflow easily reproducible, and results can be interpretable. Code used was design a little flexible to not required modification. It can be push in production to design a small application to help people during this sanitary crisis.

However, at the time we wrote this line, the constraint of 100 km around the house of people was given up. But the principle can be adapt to plan vacation on a specific area.

References

- **Principal Component Analysis for Dimensionality Reduction** - https://machinelearningmastery.com/principal-components-analysis-for-dimensionality-reduction-in-python/
- **Measures of Scale** - https://www.itl.nist.gov/div898/handbook/eda/section3/eda356.htm
- **Selecting the number of clusters with silhouette analysis on Kmeans clustering** - https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html
- **10 Clustering Algorithms** - https://machinelearningmastery.com/clustering-algorithms-with-python/
- **How test the performances of a clustering algorithms -** https://www.researchgate.net/post/How_can_I_test_the_performance_of_a_clustering_algorithm
- https://www.cs.ccu.edu.tw/~wylin/BA/Fusion_of_Biometrics_II.ppt
- http://eric.univ-lyon2.fr/~ricco/cours/slides/en/classif_interpretation.pdf
- https://eric.univ-lyon2.fr/~ricco/cours/didacticiels/Python/en/cah_kmeans_avec_python.pdf