The Battle of neighborhoods! You have to move on ? Don't panic!

Data science as a tool for real estate rental agencies

Table of Content

1 Introduction to the Business Problem

This section provides a description of the problem and a discussion of the background.

1.1 Background

Many people move from their home country every year for different reasons, some of them because are starting a new job or a new business, others for a semester of study and other for love.

All these people have something in common, they are locking for a place that is comparable to the current home.

In big cities such as Rome, Paris or London, with huge population of renters, it's common to use a real estate agent to find a rental property.

The mainly requests that real estate agency receive from customers are :

- find a house in a neighborhood that is as similar as possible to the one they come from;
- That the new neighborhood meets a list of requirements such as parks, traditional restaurants, and so on.

The aim of this work is to demonstrate how using some data science techniques it is possible to help real estate agencies to find apartments for rent that meet the needs of customers.

1.2 Problem description

A family is moving from their hometown in Rome to Paris. They ask a real estate agency to find an apartment for rent that is in a neighborhood similar to the one they are leaving and that has parks where they can walk their dog.

They would like to find a neighborhood with many restaurants and would like to be able to choose where to train between the various gyms. They would also like to have some grocery stores nearby, so they can buy the ingredients needed to cook the Italian dishes.

Summarized, the family like to have the following venues nearby:

- park;
- gym;
- restaurants & bars;
- grocery store.

And that the apartment has:

- Low price per m²;
- boroughs that is similar to the one they are currently living in.

2 Data

This section provides a description of the data and how it will be used to solve the problem.

2.1 Description of the Data

The following data will be used :

1. **Average cost of a rental house in Paris:** This information is gathered from this webpage 'https://www.seloger.com/prix-de-l-immo/location/ile-de-france/paris.htm (https://www.seloger.com/prix-de-l-immo/location/ile-de-france/paris.htm)'. The dataset consists of the district number and the average monthly cost of a rented apartment in that district.
2. **Average burglary in the borough of Paris:** This information is gathered from this webpage 'https://www.bfmtv.com/societe/carte-delinquance-a-paris-quels-sont-les-arrondissements-ou-l-on-recense-le-plus-de-delits_AN-201910180103.html (https://www.bfmtv.com/societe/carte-delinquance-a-paris-quels-sont-les-arrondissements-ou-l-on-recense-le-plus-de-delits_AN-201910180103.html)'. The dataset is composed of the district number and the number of annual burglaries in that district.
3. **Information about the venues in Paris neighboroods :** This information is gathered through FourSquare API. The dataset contains Paris neighborhood information. It consists of the district number, the neighborhood name and all the premises that are present within a 750 meter radius from the neighborhood center.
4. **Information about the venues in home town neighborood :** This information is gathered through FourSquare API. The dataset contains home town neighborhood information. It consists of the district number, the neighborhood name and all the premises that are present within a 750 meter radius from the neighborhood center.
5. **The names of all Paris neighboroods :** This information is gathered from this webpage 'https://opendata.paris.fr/explore/dataset/quartier_paris (https://opendata.paris.fr/explore/dataset/quartier_paris)'.

Not all the data is in the proper format and it needs to be transformed. The Geocoder Python package (https://geocoder.readthedocs.io/index.html (https://geocoder.readthedocs.io/index.html)) will be used to receive the latitude and logitude coordinates of all neighborhoods. The neighborhoods and their corresponding latitude and longitude will be used as input for FourSquare to get information about them

2.2 How the data will be used to solve the problem

First we will analyze the distribution of venues in the Paris neighborhoods to find those neighborhoods that best suit the preferences of the family.

Next, we'll divide the neighborhoods of Paris into clusters to find the ones that are as similar as possible to the neighborhood of the family's hometown. One hot encoding and k-means will be used for this porpouse.

The last step is to use the average rental cost per square meter and the crime rate to create a ranking of neighborhoods that meet the customer's needs.

2.3 Data Preparation

2.3.1 Import Paris boroughs dataset

Paris has in total 20 boroughs (called arrondissements in French) and are divided in 80 neighborhoods.

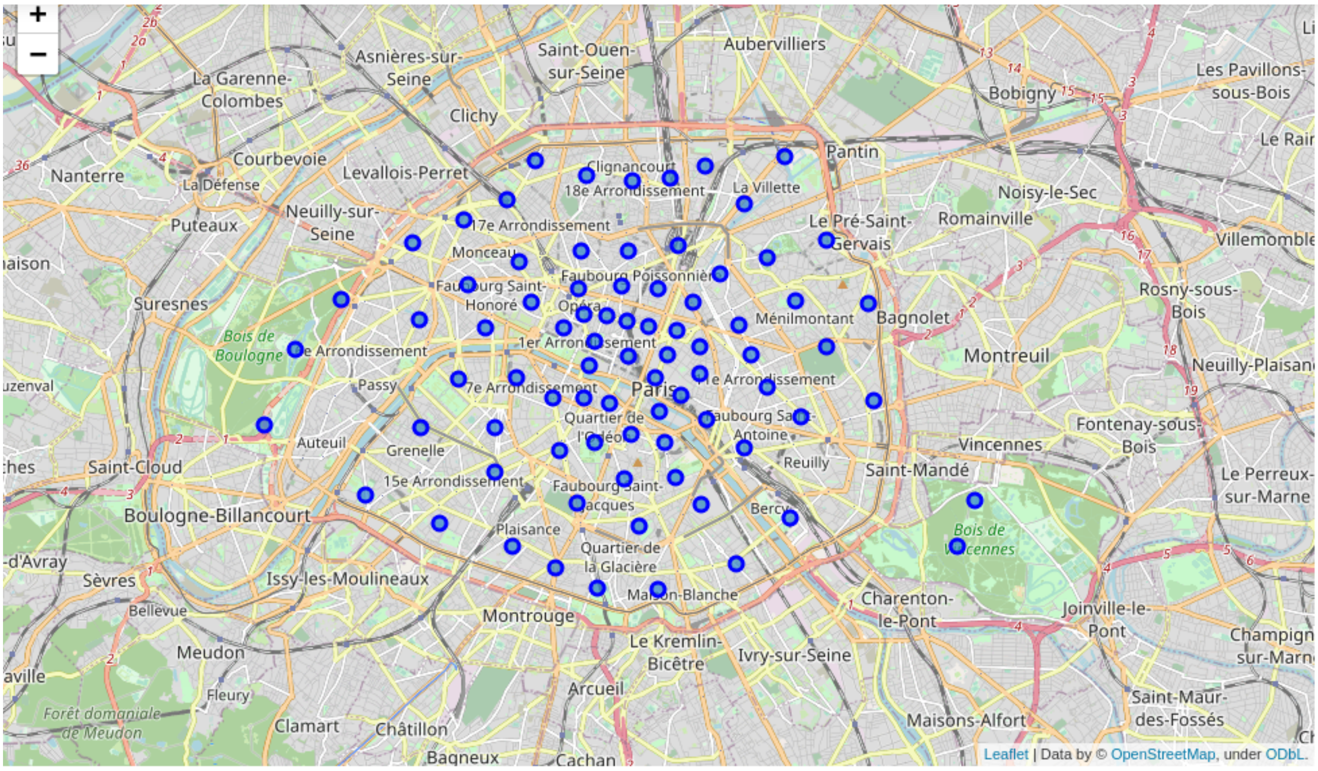
The dataset of Paris boroughs can be found at the following link:

https://opendata.paris.fr/explore/dataset/quartier_paris (https://opendata.paris.fr/explore/dataset/quartier_paris)

After rearraging data we get the following dataset (the first 5 rows)

	Neighborhood	Borough	Latitude	Longitude
0	Saint-Gervais	4	48.8557186509	2.35816233385
1	Saint-Thomas-d'Aquin	7	48.8552632694	2.32558765258
2	Porte-Saint-Denis	10	48.873617661	2.35228289495
3	Saint-Germain-l'Auxerrois	1	48.8606501352	2.33491032928
4	Villette	19	48.8876610888	2.37446821213

Here is the map of Paris and superimposed the Neighborhoods.



2.3.2 Create Paris venues dataset

Using the Foursquare API we prepare and populate a dataset that will describe each district of Paris in terms of venues.

Let's take a look at the data

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Id	Venue Category
0	Quinze-Vingts	48.8469159441	2.37440162648	Promenade plantée – La Coulée Verte	48.847632	2.375107	4bf58dd8d48988d159941735	Trail
1	Quinze-Vingts	48.8469159441	2.37440162648	Les Embruns	48.847100	2.371883	52e81612bcbc57f1066b79f2	Creperie
2	Quinze-Vingts	48.8469159441	2.37440162648	Le Calbar	48.848702	2.375487	4bf58dd8d48988d11e941735	Cocktail Bar
3	Quinze-Vingts	48.8469159441	2.37440162648	Viaduc des Arts	48.848664	2.372931	4bf58dd8d48988d1df941735	Bridge
4	Quinze-Vingts	48.8469159441	2.37440162648	Rue Crémieux	48.847021	2.371110	52e81612bcbc57f1066b7a25	Pedestrian Plaza

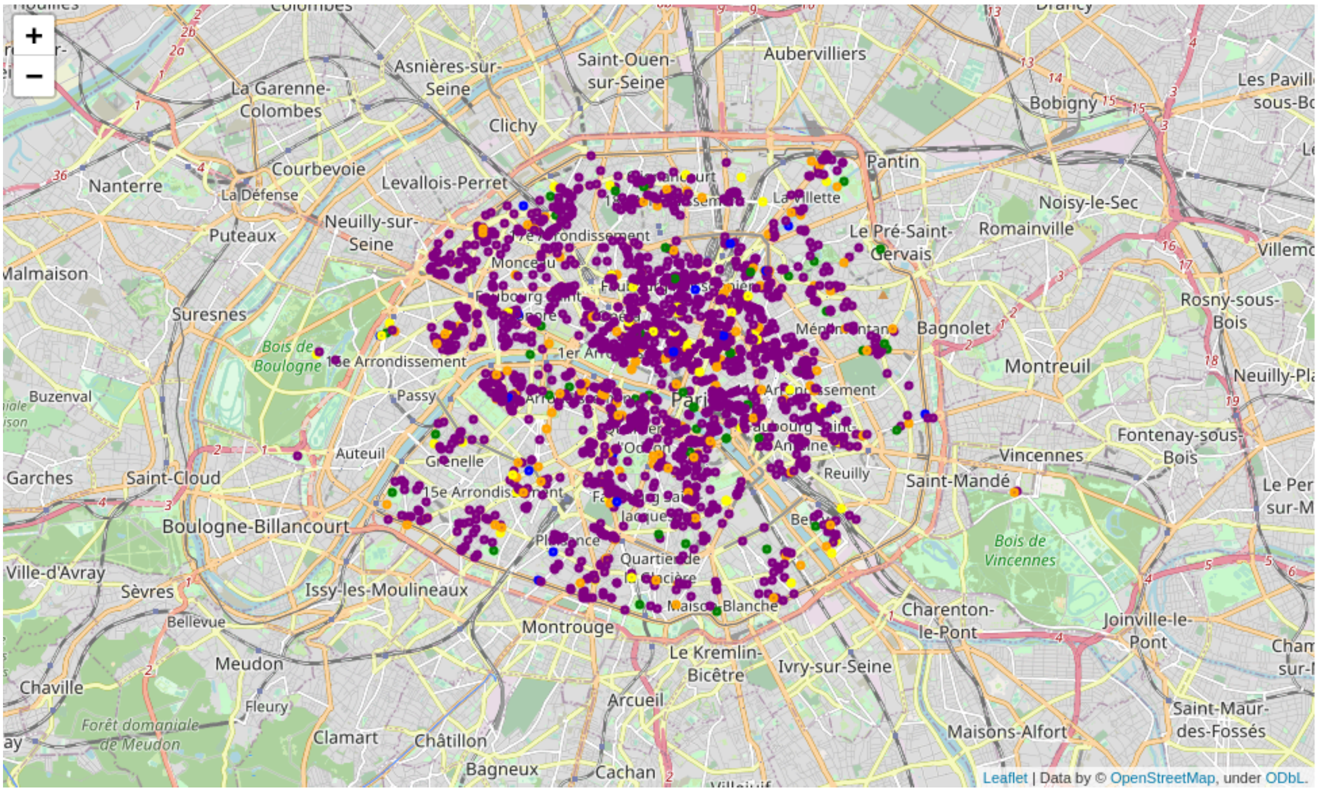
paris_venues dataset contains 5245 venues that are divided in 300 categories.

2.3.3 Create datasets about family favorite places

Starting from the paris_venues dataset we create another one that cointains the family favorite venues only.

This dataset will be used to find all neighborhoods that meet the needs of the family.

We create a map that represents the geographic distribution of favorite venues.



2.3.4 Create family hometown neighborhood dataset

Using the same steps as above we create a new dataset that describes the hometown dataset in term of venues.

We quickly check the consistency of the data.

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Id	Venue Category
0	San Paolo	41.854636	12.47997	Ilios	41.854703	12.478428	4bf58dd8d48988d10e941735	Greek Restaurant
1	San Paolo	41.854636	12.47997	Buskers Pub	41.852135	12.479969	4bf58dd8d48988d11b941735	Pub
2	San Paolo	41.854636	12.47997	Miami 3	41.851892	12.478228	4bf58dd8d48988d1c9941735	Ice Cream Shop
3	San Paolo	41.854636	12.47997	Bar San Paolo	41.856290	12.478663	4bf58dd8d48988d16d941735	Café
4	San Paolo	41.854636	12.47997	La Muffineria	41.853127	12.476754	4bf58dd8d48988d1bc941735	Cupcake Shop

2.3.5 Create average cost dataset

From '[https://www.seloger.com/prix-de-l-immo/location/ile-de-france/paris.htm (https://www.seloger.com/prix-de-l-immo/location/ile-de-france/paris.htm)](https://www.seloger.com/prix-de-l-immo/location/ile-de-france/paris.htm)' we create a simple table that contains the id of the boroughs and the average cost of a rent per square meter.

In [29]:
```
df_average_cost.head()
```

Out[29]:

	Borough	Cost
0	1	37.9
1	2	36.9
2	3	37.3
3	4	38.6
4	5	36.3

2.3.6 Create burglary per year dataset

From https://www.bfmtv.com/societe/carte-delinquance-a-paris-quels-sont-les-arrondissements-ou-l-on-recense-le-plus-de-delits_AN-201910180103.html (https://www.bfmtv.com/societe/carte-delinquance-a-paris-quels-sont-les-arrondissements-ou-l-on-recense-le-plus-de-delits_AN-201910180103.html) we create a simple table that contains the id of the boroughs and number of burglary per year.

In [31]:
```
df_burglary_year.head()
```

Out[31]:

	Borough	Burglary
0	1	302
1	2	516
2	3	446
3	4	396
4	5	435

3 Methodology

This is the principal part of the work.

We start analyzing Paris venues in order to find the list of neighborhoods that meets family requirements.

3.1 Neighborhoods that meets family requirements

Rearrange the favorite venues dataset to present the data in a different way .

```
favorite_venues_grouped = favorite_venues.groupby('Neighborhood')['Venue Category'].value_counts().unstack().fillna(
favorite_venues_grouped.head()
```

Venue Category	Café	Grocery	Gym	Park	Restaurant
Neighborhood					
Amérique	1.0	0.0	0.0	1.0	2.0
Archives	1.0	0.0	0.0	0.0	28.0
Arsenal	1.0	0.0	1.0	3.0	29.0
Arts-et-Métiers	1.0	1.0	0.0	1.0	40.0
Auteuil	0.0	0.0	0.0	0.0	1.0

Not all neighborhoods satisfy all family needs, we only select those that satisfy all of them.

```
favorite_venues_grouped[(favorite_venues_grouped[['Gym','Park', 'Café', 'Grocery', 'Restaurant']] != 0).all(axis=1)]
```

Venue Category / Neighborhood	Café	Grocery	Gym	Park	Restaurant
Batignolles	3.0	1.0	1.0	2.0	48.0
Hôpital-Saint-Louis	4.0	1.0	1.0	1.0	43.0
Palais-Royal	3.0	1.0	1.0	1.0	34.0
Porte-Dauphine	1.0	1.0	2.0	1.0	1.0

Only four neighborhoods meet all the needs of the family.

3.2 Neighborhoods similar to the one of the hometown

For finding neighborhoods similar to that of the hometown we use k means clustering. k means clustering is an unsupervised machine learning algorithm that is able to partitioning a dataset into groups of elements that have similar characteristics. in our case we want to group the neighborhoods according to the distribution of the venues.

3.2.1 Preparing data for clustering

We create a dataset that contains all the neighborhoods and venues of Paris and the venues or Rome neighborhoods.

```
mixed_neighborhoods.head()
```

	Neighborhood	Borough	Latitude	Longitude
0	Saint-Gervais	4	48.8557186509	2.35816233385
1	Saint-Thomas-d'Aquin	7	48.8552632694	2.32558765258
2	Porte-Saint-Denis	10	48.873617661	2.35228289495
3	Saint-Germain-l'Auxerrois	1	48.8606501352	2.33491032928
4	Villette	19	48.8876610888	2.37446821213

We set Borough to 0 for Rome neighborhood.

In [36]:
```
home_neighborhood = {'Neighborhood':'San Paolo', 'Borough':0, 'Latitude': latitude_rome, 'Longitude':longitude_rome}
mixed_neighborhoods = mixed_neighborhoods.append(home_neighborhood, ignore_index=True)
```

In [37]:
```
mixed_venues = paris_venues.append(rome_nearby_venues)
```

In [38]:
```
mixed_venues.shape
```

Out[38]: (5305, 8)

```
In [39]: mixed_neighborhoods.shape
```

```
Out[39]: (81, 4)
```

For applying the k means clustering algorithm we have to transform all the categorical variables. The one hot encoding tecnique will be used.

```
# one hot encoding
cluster_onehot = pd.get_dummies(mixed_venues[['Venue Category']], prefix="", prefix_sep="")
cluster_onehot.head()
```

	Accessories Store	Afghan Restaurant	African Restaurant	Alsatian Restaurant	American Restaurant	Arcade	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Art Museum	...	Vegetarian / Vegan Restaurant	Venezuelan Restaurant	Video Game Store	Vietna Resta
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	

5 rows × 303 columns

We explore the one hot encoding dataset.

The top ten venues per neighborhood are

```
neighborhoods_venues_sorted.head()
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Amérique	Plaza	French Restaurant	Supermarket	Pool	Bed & Breakfast	Park	Café	Theater	Bistro	Zoo Exhibit
1	Archives	French Restaurant	Clothing Store	Coffee Shop	Bistro	Hotel	Art Gallery	Plaza	Bookstore	Burger Joint	Cocktail Bar
2	Arsenal	French Restaurant	Hotel	Plaza	Park	Tapas Restaurant	Boat or Ferry	Seafood Restaurant	Thai Restaurant	Cocktail Bar	Pedestrian Plaza
3	Arts-et-Métiers	French Restaurant	Hotel	Cocktail Bar	Italian Restaurant	Wine Bar	Bar	Vietnamese Restaurant	Restaurant	Chinese Restaurant	Coffee Shop
4	Auteuil	Tennis Court	Stadium	Garden	Outdoors & Recreation	French Restaurant	Racecourse	Sporting Goods Shop	Plaza	Museum	Botanical Garden

```
neighborhoods_venues_sorted[neighborhoods_venues_sorted['Neighborhood'] == 'San Paolo']
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
75	San Paolo	Italian Restaurant	Café	Pizza Place	Ice Cream Shop	Park	Pub	Fast Food Restaurant	Asian Restaurant	Clothing Store	Bistro

I'm from Rome and I know quite well San Paolo neighborhood. Since the district became the seat of the third university of Rome, many restaurants, pubs and fast food have been opened. The data we obtained from FourSquare API correctly represent the distribution of the venues in San Paolo.

3.2.2 Clustering

Now everything is ready for clustering, let's see what happen.

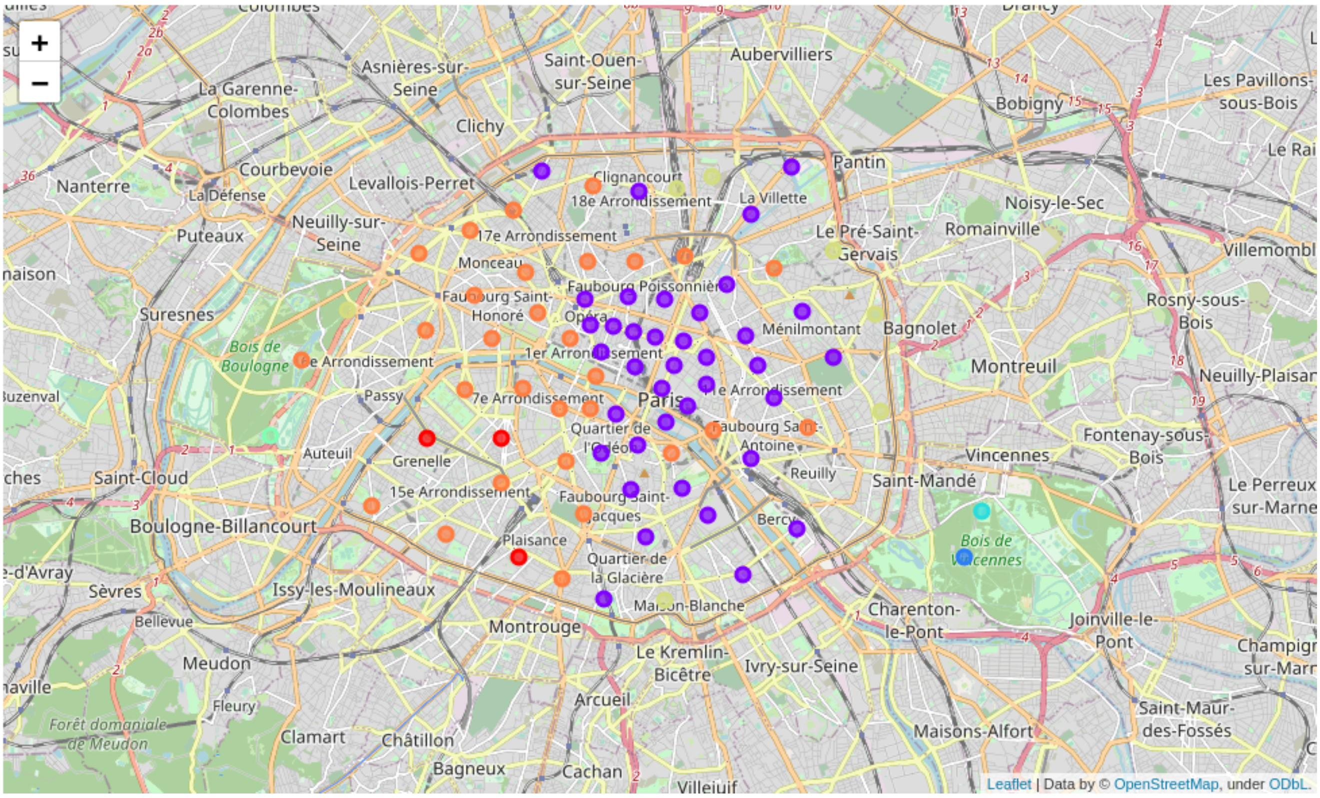
In [47]:
```
# set number of clusters
kclusters = 7

cluster_grouped_clustering = cluster_grouped.drop('Neighborhood',
1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(cluster
_grouped_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```

Out[47]:
```
array([5, 1, 6, 1, 4, 6, 3, 1, 1, 1], dtype=int32)
```

```
cluster_merged.head() # check the last columns!
```

	Neighborhood	Borough	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
0	Saint-Gervais	4	48.8557186509	2.35816233385	1	French Restaurant	Clothing Store	Italian Restaurant	Hotel	Ice Cream Shop	Gay Bar	Thai Restaurant	Gourmet Shop
1	Saint-Thomas-d'Aquin	7	48.8552632694	2.32558765258	6	French Restaurant	Hotel	Café	Art Gallery	Coffee Shop	Italian Restaurant	American Restaurant	Historic Site
2	Porte-Saint-Denis	10	48.873617661	2.35228289495	1	Hotel	French Restaurant	Bakery	Bar	Bistro	Vegetarian / Vegan Restaurant	Vietnamese Restaurant	Indian Restaurant
3	Saint-Germain-l'Auxerrois	1	48.8606501352	2.33491032928	6	French Restaurant	Hotel	Plaza	Coffee Shop	Art Museum	Historic Site	Bar	Italian Restaurant
4	Villette	19	48.8876610888	2.37446821213	1	Hotel	Bar	French Restaurant	Café	Asian Restaurant	Food Truck	Fast Food Restaurant	Multiplex

Let's show in a map the geographic cluster distribution.



Let's see in which cluster San Paolo is located

```
cluster_merged.loc[cluster_merged['Borough'] == 0]
```

	Neighborhood	Borough	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	Co
80	San Paolo	0	41.8546	12.48	1	Italian Restaurant	Café	Pizza Place	Ice Cream Shop	Park	Pub	Fast Food Restaurant	Asian Restaurant	Clothing Store	

The hometown neighborhood belongs to cluster 1.

And here is the list of all Paris neighborhood in cluster 1. There are 38 neighborhoods in the same cluster, only the first 5 are showed.

```
cluster_merged.loc[cluster_merged['Cluster Labels'] == 1]
```

	Neighborhood	Borough	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	
0	Saint-Gervais	4	48.8557186509	2.35816233385	1	French Restaurant	Clothing Store	Italian Restaurant	Hotel	Ice Cream Shop	Gay Bar	Thai Restaurant	
2	Porte-Saint-Denis	10	48.873617661	2.35228289495	1	Hotel	French Restaurant	Bakery	Bar	Bistro	Vegetarian / Vegan Restaurant	Vietnamese Restaurant	F
4	Villette	19	48.8876610888	2.37446821213	1	Hotel	Bar	French Restaurant	Café	Asian Restaurant	Food Truck	Fast Food Restaurant	
5	Quinze-Vingts	12	48.8469159441	2.37440162648	1	French Restaurant	Coffee Shop	Sandwich Place	Hotel	Bakery	Bar	Farmers Market	Tra
7	Bercy	12	48.8352090499	2.38621008421	1	Hotel	Italian Restaurant	Bus Stop	Bakery	Gym / Fitness Center	French Restaurant	Wine Bar	

There are only two neighborhoods that are shared with cluster 1 and family needs:

- Hôpital-Saint-Louis
- Palais-Royal

3.3 Average cost and burglary rate

From the analysis of the sites we have identified two neighborhoods that meet all customer requirements:

- Hôpital-Saint-Louis
- Palais-Royal

Now let's see what are the average rental cost and the burglary rate in these two neighborhoods

```
neighborhoods.loc[(neighborhoods['Neighborhood'] == 'Hôpital-Saint-Louis') | (neighborhoods['Neighborhood'] == 'Pala
```

	Neighborhood	Borough	Latitude	Longitude
34	Hôpital-Saint-Louis	10	48.87600829	2.36812301789
52	Palais-Royal	1	48.8646599781	2.33630891897

```
df_average_cost.loc[(df_average_cost['Borough'] == 1) | (df_average_cost['Borough'] == 10)]
```

	Borough	Cost
0	1	37.9
9	10	32.3

```
df_burglary_year.loc[(df_burglary_year['Borough'] == 1) | (df_burglary_year['Borough'] == 10)]
```

	Borough	Burglary
0	1	302
9	10	790

4 Results

We found four neighborhoods that had all the features the customer requested. Using the k-means clustering algorithm we found 38 neighborhoods that are similar to customer hometown neighborhood. The intersection of the two previous results gives only two neighborhoods.

Using the information from cost and crime rate we can summarize the result in the following table :

Neighborhood	Cost per sqm	Burglary Rate
Hôpital-Saint-Louis	32.3	790
Palais-Royal	37.9	302

Considering a 100 square meter apartment, the difference in rent is 50 euros and the risk of burglary is reduced by half.

Anyway we left the choise to the customer.

5 Discussion

We have use the simplest clustering algorithm, one can try to use other clustering algorithms and find which one is best for this type of problem.

Other clustering algorithm can be used in order to find the best for this kind of problem.

Moreover, having a customer history, one could think of creating user profiles to use with recommendation system.

6 Conclusion

The aim of this project was to identify a neighborhood similar to the client's current one and which, at the same time, also had venues that were important to him.

We have succeeded in demonstrating that data science methodologies can be used for the solution of this type of problem.

As a future development, the use of recommendation systems could be investigated to get further information on choosing the apartment to rent.