Paris Boroughs Cluster Classification By Venues Using K-means

Manuel Benedicto

20 March 2020

1. Introduction

For many people, the search for an excellent area to live is an arduous process of investigation into the different areas of one city. The definition of a good area to live into varies from one person to another and so the search for a house or an apartment is in the same way unique to the individual.

In this project we will try to find an optimal neighborhood for a fictional client. This client will have some criteria that will define their ideal area to live in Paris (France). For the purpose of simplicity we are going to take only three elements into account for this fictional user:

- The apartment should be near of a pharmacy
- It should also have a supermarket nearby
- And it should not be far from a metro or train station

2. Data compilation

We will need to retrieve the data needed for our analysis of different sources and afterwards we are going to apply some data cleaning to our datasets as they may contain some information we are not going to need to perform our analysis.

2.1 Data sources

On one hand, we will get the shape and characteristics of all boroughs in the city of Paris from the official website of French government Open Platform for French Public Data (https://www.data.gouv.fr/en/datasets/quartiers-administratifs/). From this website we get this dataset:

```
boroughs_url = 'https://www.data.gouv.fr/es/datasets/r/a8748f53-5850-4a04-b8cc-9c9f5f72949f'
boroughs = gpd.read_file(boroughs_url)
boroughs.head()
```

Out[1]:

	n_sq_qu	n_sq_ar	c_qu	surface	l_qu	perimetre	c_quinsee	c_ar	geometry
0	750000048	750000012	48	1.235916e+06	Quinze-Vingts	4509.486974	7511204	12	POLYGON ((2.37320 48.84057, 2.37241 48.84017, ...
1	750000007	750000002	7	2.781426e+05	Mail	2179.153605	7510203	2	POLYGON ((2.34684 48.86491, 2.34668 48.86443, ...
2	750000008	750000002	8	2.814482e+05	Bonne-Nouvelle	2233.976030	7510204	2	POLYGON ((2.35152 48.86443, 2.35095 48.86341, ...
3	750000050	750000013	50	3.044178e+06	Gare	7070.350567	7511302	13	POLYGON ((2.36771 48.81742, 2.36696 48.81719, ...
4	750000070	750000018	70	1.653715e+06	Clignancourt	6005.520389	7511802	18	POLYGON ((2.35168 48.89139, 2.35145 48.89043, ...

Figure 1. Preview of borough's dataset

This dataset gives us an overview of how the boroughs in Paris are shaped. In fact, they are shaped as polygons and there are 80 of them. In order to be able to work better on this data we are going to add centroids to each borough which will serve us to look for venues around them.

```
boroughs = gpd.GeoDataFrame.from_features(boroughs)
boroughs['centroid_lon'] = boroughs['geometry'].centroid.x
boroughs['centroid_lat'] = boroughs['geometry'].centroid.y
boroughs = boroughs.sort_values(by=['l_qu'])
boroughs.crs = {'init': 'epsg:4326'}
boroughs.to_csv(path_or_buf='boroughs.csv')
pd.read_csv('boroughs.csv')
```

Out[2]:

geometry	c_ar	c_qu	c_quinsee	l_qu	n_sq_ar	n_sq_qu	perimetre	surface	centroid_lon	centroid_lat
POLYGON 2.409402172235365 .88019204178156,...	19	75	7511903	Amérique	750000019	750000075	6399.022082	1.835720e+06	2.395440	48.881638
POLYGON 2.368479720528894 .85583081045625,...	3	11	7510303	Archives	750000003	750000011	2534.100042	3.677284e+05	2.363205	48.859192
POLYGON 2.368512371393433 .85573412813671,...	4	15	7510403	Arsenal	750000004	750000015	2878.559656	4.872649e+05	2.364768	48.851585
POLYGON 2.360209979547445 .86519024025307,...	3	9	7510301	Arts-et-Métiers	750000003	750000009	2482.460453	3.180877e+05	2.357083	48.866470
POLYGON 2.249224929777843 .85782761493475,...	16	61	7511601	Auteuil	750000016	750000061	12452.253931	6.383888e+06	2.252277	48.850622
...	...	...	...	...	...	...	...	...	...	...
POLYGON 2.349244542106854 .84451631667142,...	5	20	7510504	Sorbonne	750000005	750000020	2892.944068	4.331978e+05	2.345747	48.849045
POLYGON 2.295039618663717 .87377869547586,...	17	65	7511701	Ternes	750000017	750000065	5264.597082	1.465071e+06	2.289964	48.881178

Figure 2. New dataset with centroids of each borough

On the other hand, we are going to collect the data required for our analysis: location of every pharmacy, supermarket and metro station within every quartier in the city of Paris. To get this data we are going to use the Yelp's API.

	Unnamed: 0	Borough	Borough Latitude	Borough Longitude	Venue Name	Venue Category	Venue Latitude	Venue Longitude	Venue City
0	0	Amérique	48.881638	2.395440	Aux deux mille-pates	Grocery	48.875530	2.391130	Paris
1	1	Amérique	48.881638	2.395440	E. Leclerc	Grocery	48.891055	2.403272	Pantin
2	2	Amérique	48.881638	2.395440	Carrefour City	Grocery	48.882940	2.394190	Paris
3	3	Amérique	48.881638	2.395440	Lidl	Grocery	48.879120	2.392570	Paris
4	4	Amérique	48.881638	2.395440	G20	Grocery	48.885903	2.394181	Paris
...	...	...	...	...	...	...	...	...	...
2889	2889	Vivienne	48.869100	2.339461	Pharmacie Moderne de Paris	Pharmacy	48.866711	2.347130	Paris
2890	2890	Vivienne	48.869100	2.339461	Pharmacie des Martyrs	Pharmacy	48.876970	2.339370	Paris
2891	2891	Vivienne	48.869100	2.339461	Marché de l'Opéra	Grocery	48.869835	2.330854	Paris
2892	2892	Vivienne	48.869100	2.339461	Bendavid Ouyoussef Goldfarb Marie	Pharmacy	48.876700	2.341600	Paris
2893	2893	Vivienne	48.869100	2.339461	Pharmacie de l'Opéra Mogador	Pharmacy	48.874050	2.331290	Paris

2894 rows × 9 columns

Figure 3. Dataset created using Yelp's API

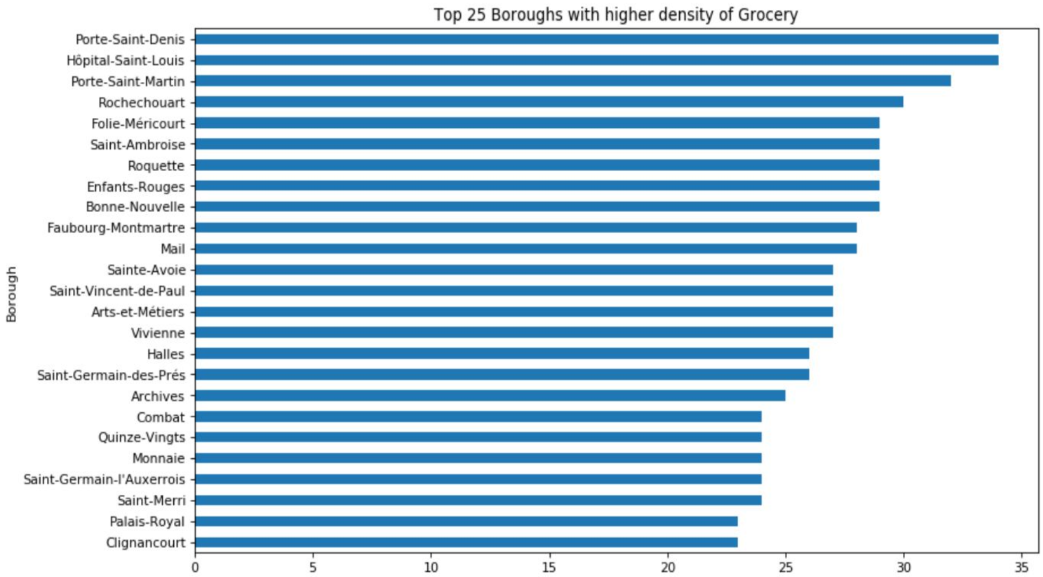
2.2 Data preprocessing and cleaning

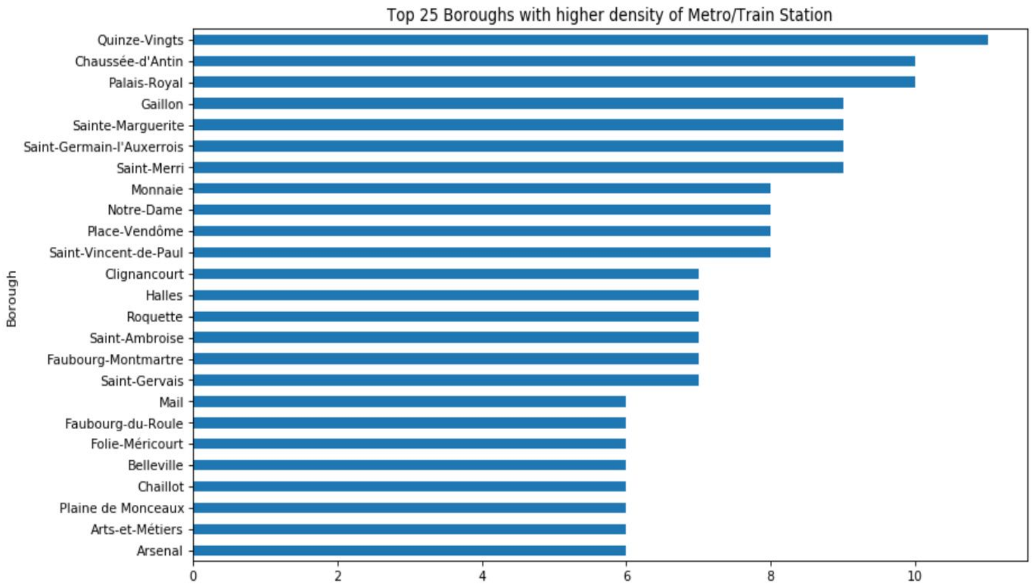
The dataset we are using for our analysis is slightly large and might contain some null values or data we are not interested in. We are going to be sure that only the venues retrieved are within the limits of Paris and we are going to also check that they belong to the categories 'Grocery', 'Pharmacy' and 'Metro Station'.

For the sake of simplicity we are going to take into account those three categories alongside 'Train Station' and we are going to ignore variants of those like 'Bakeries'. We are also going to merge the categories 'Metro Station' and 'Train Station' into the 'Metro/Train Station' category. We are also going to delete all null values in our dataset if there is any.

3. Exploratory Data Analysis

First of all we have created a new dataframe containing the total number of each type of venue in order to plot the data into some graphics to have an idea of the boroughs with higher density in each type of venue. These are the first graphics we have created using matplotlib library.


Top 25 Boroughs with higher density of Grocery
Porte-Saint-Denis
Hôpital-Saint-Louis
Porte-Saint-Martin
Rochechouart
Folie-Méricourt
Saint-Ambroise
Roquette
Enfants-Rouges
Bonne-Nouvelle
Faubourg-Montmartre
Mail
Sainte-Avoie
Saint-Vincent-de-Paul
Arts-et-Métiers
Vivienne
Halles
Saint-Germain-des-Prés
Archives
Combat
Quinze-Vingts
Monnaie
Saint-Germain-l'Auxerrois
Saint-Merri
Palais-Royal
Clignancourt
Borough
0
5
10
15
20
25
30
35


Top 25 Boroughs with higher density of Metro/Train Station
Quinze-Vingts
Chaussée-d'Antin
Palais-Royal
Gaillon
Sainte-Marguerite
Saint-Germain-l'Auxerrois
Saint-Merri
Monnaie
Notre-Dame
Place-Vendôme
Saint-Vincent-de-Paul
Clignancourt
Halles
Roquette
Saint-Ambroise
Faubourg-Montmartre
Saint-Gervais
Mail
Faubourg-du-Roule
Folie-Méricourt
Belleville
Chaillot
Plaine de Monceaux
Arts-et-Métiers
Arsenal
Borough
0
2
4
6
8
10

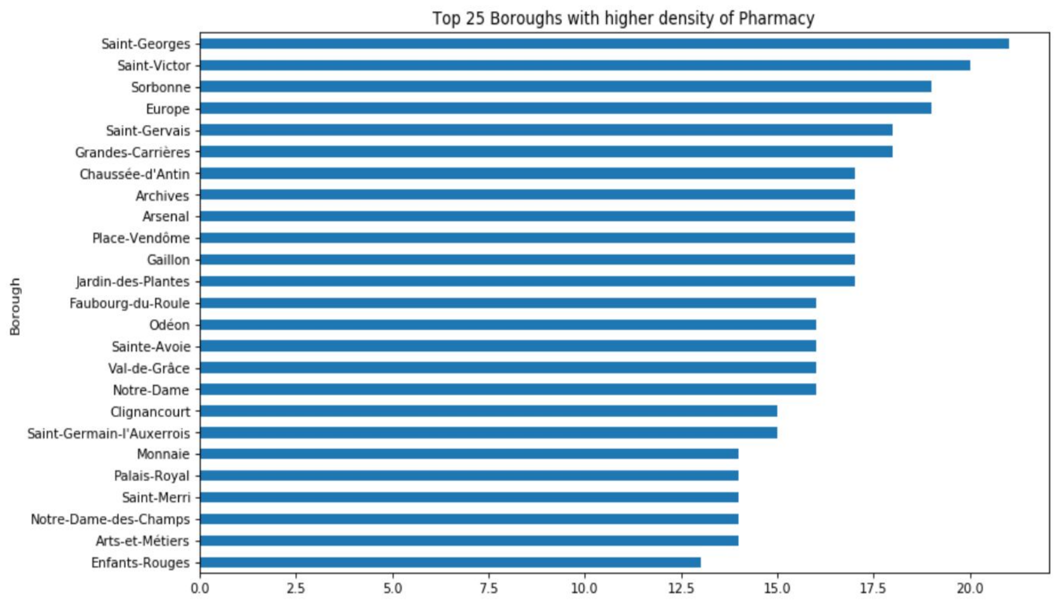


Figure 4, 5 & 6. Top boroughs in high density of each type of venue

We are now able to visualize the location of each type of venue in a map along with the centroid of each borough so that in a brief look we can figure out what areas fit our search criteria.

- Centroids of each venue are coloured in black.
- Groceries are coloured in yellow.
- Pharmacies are coloured in green.
- Metro and train stations are coloured in red.

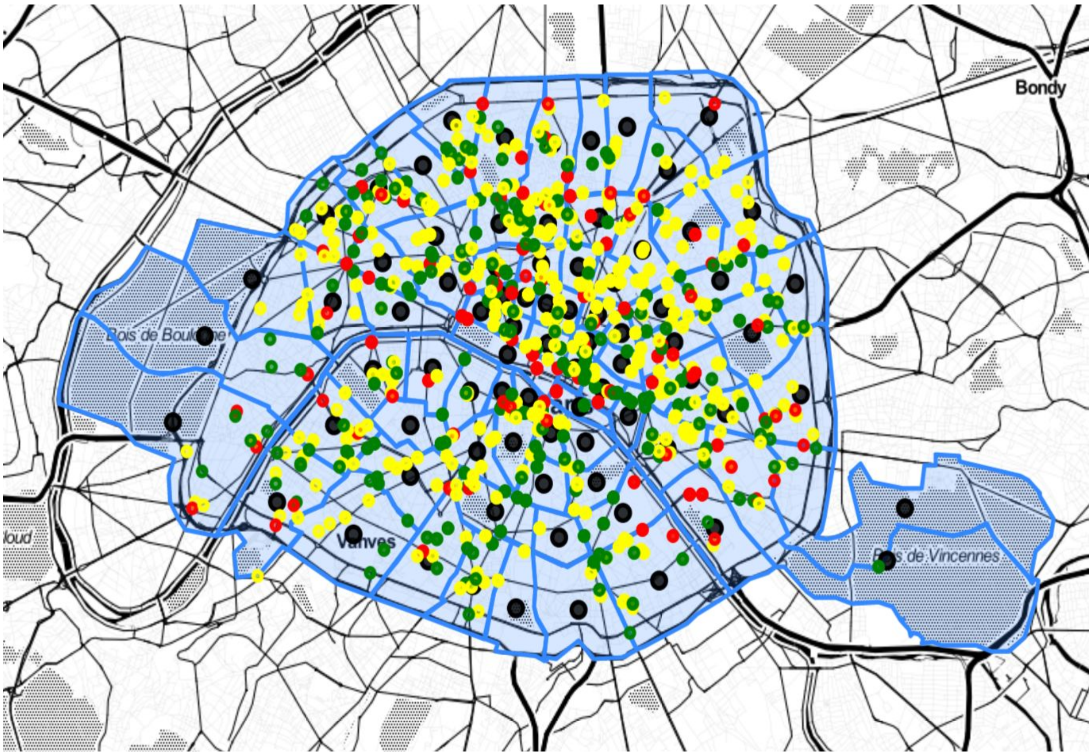


Figure 7. Map of venues

This visualization helps us formulate a hypothesis about the ideal boroughs for our client. We can say that maybe the central and northern neighbors would be the best following the criteria we had to do our search.

4. Clustering

We are going to create clusters using k-means clustering to identify boroughs that are most populated with the venues we have picked as our criteria to find the best borough to live in.

We'll be using k-means clustering for our analysis. These were preliminary results with different number of clusters:

- 2 clusters only show the uptown/downtown divide of the boroughs
- 3 clusters give more accuracy to our model but is not divided enough
- 4 clusters also identify neighborhoods with very low density of venues and gives to our model more accuracy
- 5 clusters and more create more groups than needed for our general analysis

For this data analysis we are going to use 4 clusters as we think is the number that might fit the most our dataset and the results we are looking for.

With a boxplot we are going to see which cluster is the one that is more crowded with those venues and finally we are going to see which boroughs belong to each cluster and how many venues those boroughs have in a new map.

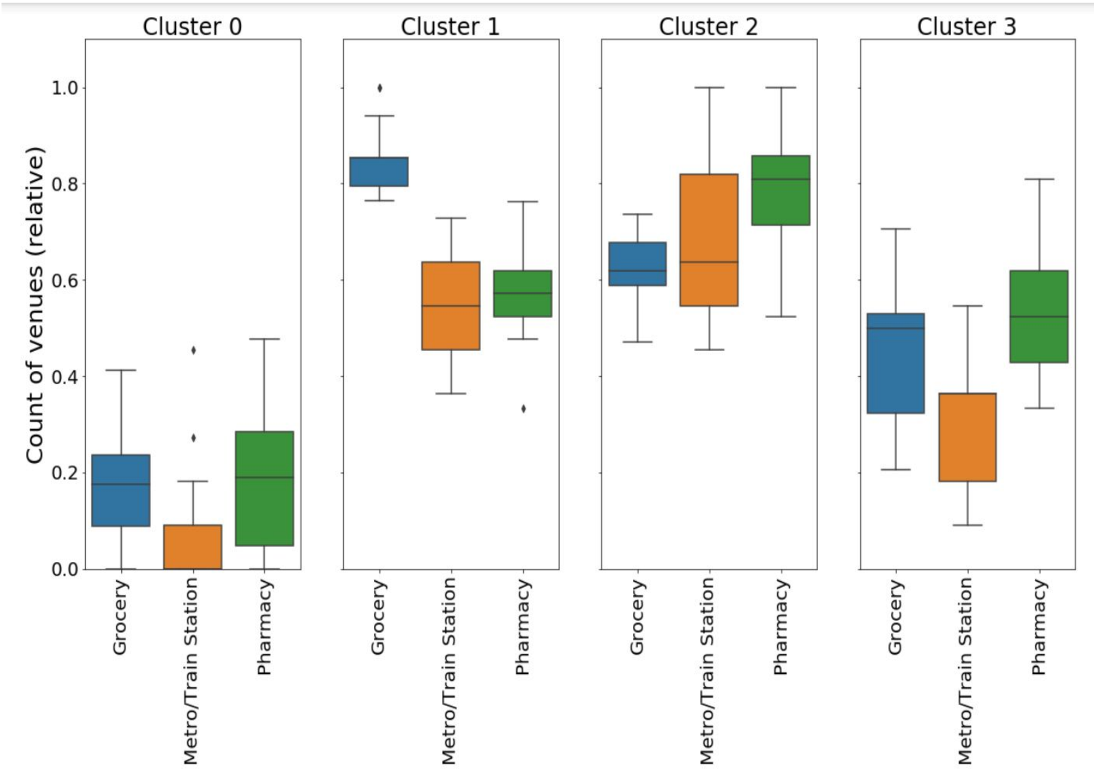


Figure 8. Clusters created using K-means

As we can see in our boxplot the cluster of boroughs with higher density of the venues defined by our criteria is cluster number 2 and is followed by the cluster number 1.

We are going to render the clusters into a map using Folium and replace the centroids of the boroughs with a new set of colors depending of the cluster they are part of.

- Cluster number 0 is represented by red
- Cluster number 1 is represented by orange
- Cluster number 2 is represented by green
- Cluster number 3 is represented by blue

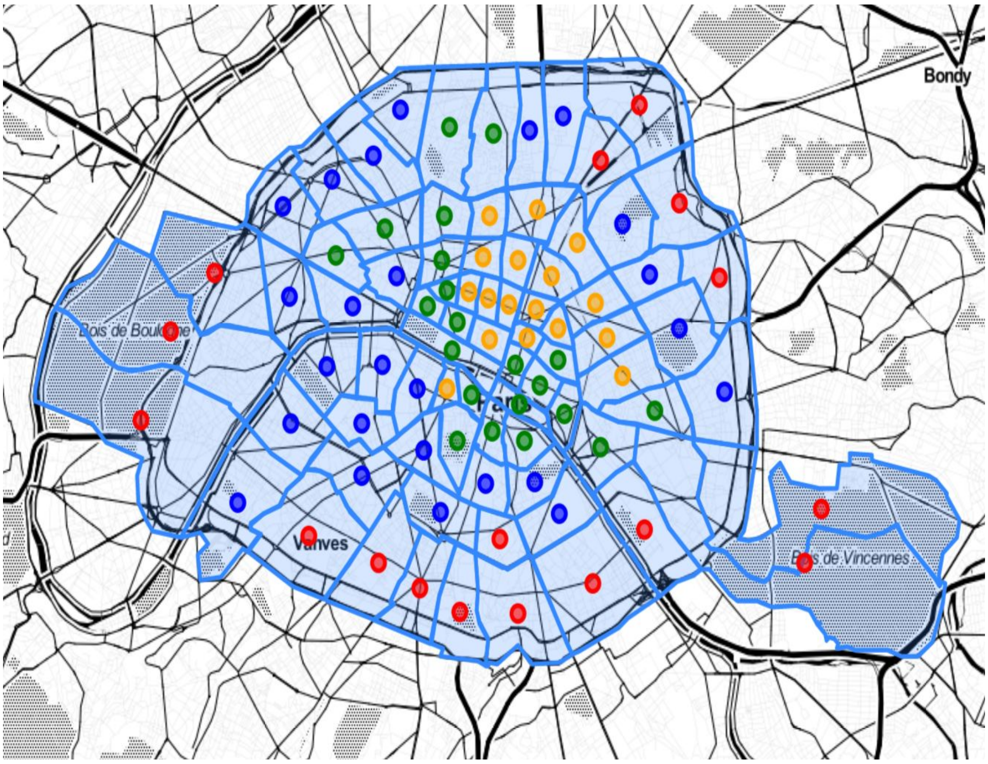


Figure 9. Map of cluster classification of boroughs

5. Conclusions

The results in the map shown at the end of our analysis section confirm our first hypothesis after looking at the map representing with markers all the venues.

In our analysis we were looking for boroughs in Paris that had the highest number of groceries, pharmacies and metro and train stations because our client wanted to find an area to live in where those three venues were nearby.

The central and northern boroughs seem to be the most crowded with the venues we picked for our analysis due to probably the higher development and higher population in the past and nowadays due to tourism as well for the central boroughs and the higher density of population nowadays in the north-western boroughs.

It would be interesting to look for a validation of our hypothesis of correlation between our results and the score of population or number of tourist attendance in each of the parisian boroughs.