# Restaurants in Madrid

This is the report for the final project of IBM's Applied Data Science capstone in Coursera. During the course, we used python and the Foursquare API to explore the cities of New York and Toronto. This project will use these tools to explore the neighborhoods in the city of Madrid, Spain. We will use open data sources and the Foursquare API and try to get an idea of the characteristics of the neighborhoods in Madrid.

## 1. The Problem and the Approach

Madrid is a big city, with more than 3M inhabitants, and comprises 21 districts and 131 neighborhoods. Madrid is a very heterogeneous city, with very densely populated neighborhoods, touristic neighborhoods, popular neighborhoods, business areas, etc.

Let's suppose we want to open a restaurant in Madrid.We still need to decide the type of restaurant — a Spanish restaurant? or something more exotic? — and the menu — will we focus on price or go "premium"?. In order to take a decision, we will use data analysis to explore the different neighborhoods in Madrid, including aspects such as:

- What type of business are there in the neighborhood? (i.e. which is the most popular type of cuisine in the neighborhood? what is the average price of a menu in a restaurant in the neighborhood?)
- How many similar businesses are in the neighborhood?
- Who lives in the neighborhood? (e.g. is this a densely populated neighborhood, are there mostly aged population?)

We will import and analyze these data and try to decide the better place to open a restaurant.

# 2. The Data

As requested by the project, we will use the Foursquare API to get a listing recommended venues in each neighborhood. For the rest of the information, the Municipality of Madrid has an Open Data Portal where they share up-to-date data about different aspects of the city, such as number of inhabitants, their ages and other socioeconomic indicators. We will use the following data:

- Madrid neighborhoods, (CSV, 10 KB)with information about the perimeter and area.
- Catalog of venues and their activities (December 2020) (CSV, 60491 KB), organized per neighborhood
- City population census (December 2020), (CSV) with information about the number of inhabitants per age in each neighborhood.
- Neighborhoods GeoJson, with geographical information about the neighborhoods in Madrid (provided by CartoDB)

From the FourSquare API, we will get:

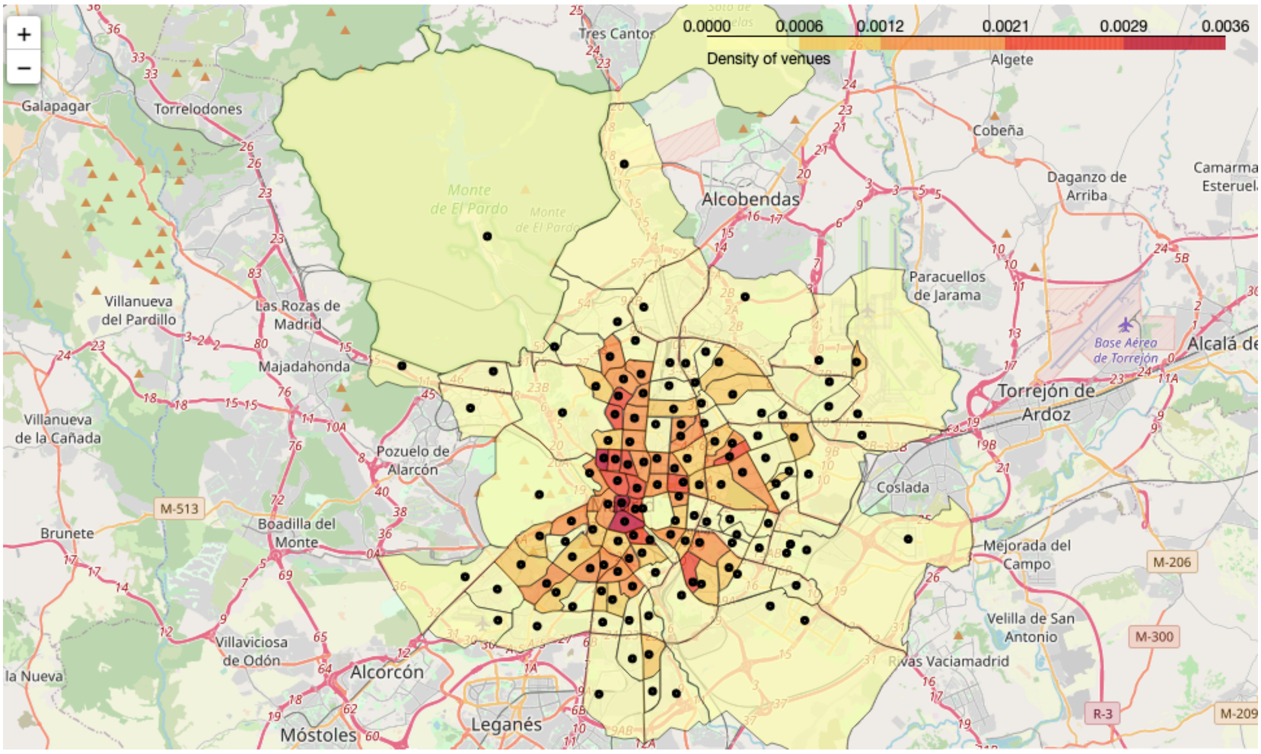
- All recommended venues around a location.
- All recommended venues around a location by price category.

# 3. The methodology

## Exploring the neighborhoods in Madrid

First of all, we will use the data downloaded from the Open City Portal of Madrid to explore the neighborhoods in Madrid. First, we used the neighborhoods dataset provided by the Municipality of Madrid to get general information about the neighborhoods, such as their name, their unique identifier, the perimeter and the area. We also had the geographical information from the .geojson file provided by CartoDB. However, in order to make the requests to the Foursquare API, we need a "centroid" for each neighborhood. We have used the Nominatim geocoder to get the coordinates for each neighborhood.:

Using the venues catalog available in the Open Data Portal, we explored the neighborhoods with the highest density of venues, as presented in the following map:



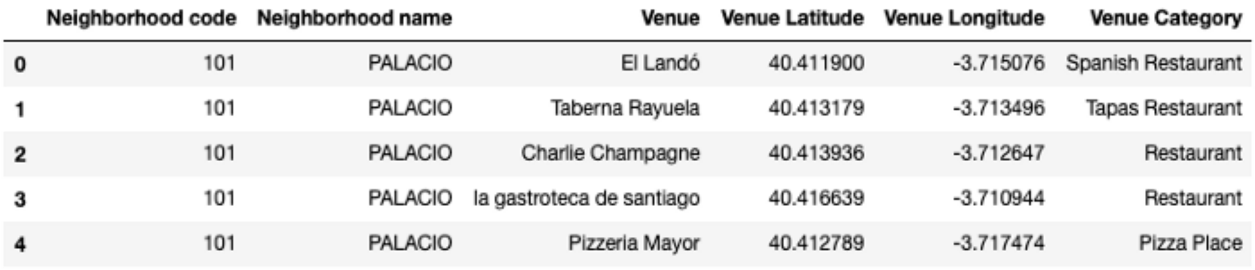
As expected, the density of venues is higher near the center of the city.

## Getting data from the Foursquare API

We will get two type of data from the FourSquare API: in both cases, we will use the Recommended Venues endpoint, that returns a list of recommended venues near a given location.
First, we will get all recommended venues in the neighborhood and classify the neighborhoods based on the category of the venues.
Then, we will request the list of recommended venues in a location for all price ranges defined in Foursquare.

## Clustering neighborhoods by venue category

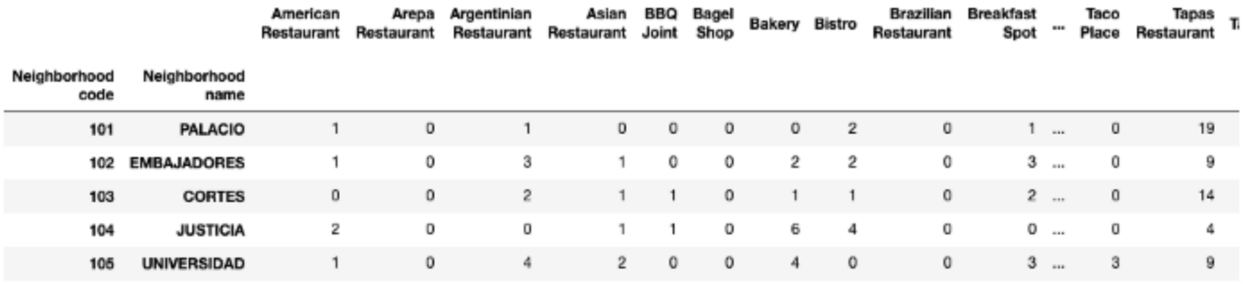
Using the Venue Recommendations endpoint, we have requested the recommended venues near the centroid of each neighborhood. With this, we got a dataframe with 11482 venues, with the following structure:



| | Neighborhood code | Neighborhood name | Venue | Venue Latitude | Venue Longitude | Venue Category |
|---|---|---|---|---|---|---|
| 0 | 101 | PALACIO | El Landó | 40.411900 | -3.715076 | Spanish Restaurant |
| 1 | 101 | PALACIO | Taberna Rayuela | 40.413179 | -3.713496 | Tapas Restaurant |
| 2 | 101 | PALACIO | Charlie Champagne | 40.413936 | -3.712647 | Restaurant |
| 3 | 101 | PALACIO | la gastroteca de santiago | 40.416639 | -3.710944 | Restaurant |
| 4 | 101 | PALACIO | Pizzeria Mayor | 40.412789 | -3.717474 | Pizza Place |

Figure 2. Dataframe of venues

Applying some transformations, we have created a dataframe with one row for each neighborhood and one column for each venue category. The values are the total number of venues of the given category in each neighborhood. The 5 first rows of the resulting dataframe are shown in the next figure:



| Neighborhood code | Neighborhood name | American Restaurant | Arepa Restaurant | Argentinian Restaurant | Asian Restaurant | BBQ Joint | Bagel Shop | Bakery | Bistro | Brazilian Restaurant | Breakfast Spot | ... | Taco Place | Tapas Restaurant | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 101 | PALACIO | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | ... | 0 | 19 | |
| 102 | EMBAJADORES | 1 | 0 | 3 | 1 | 0 | 0 | 2 | 2 | 0 | 3 | ... | 0 | 9 | |
| 103 | CORTES | 0 | 0 | 2 | 1 | 1 | 0 | 1 | 1 | 0 | 2 | ... | 0 | 14 | |
| 104 | JUSTICIA | 2 | 0 | 0 | 1 | 1 | 0 | 6 | 4 | 0 | 0 | ... | 0 | 4 | |
| 105 | UNIVERSIDAD | 1 | 0 | 4 | 2 | 0 | 0 | 4 | 0 | 0 | 3 | ... | 3 | 9 | |

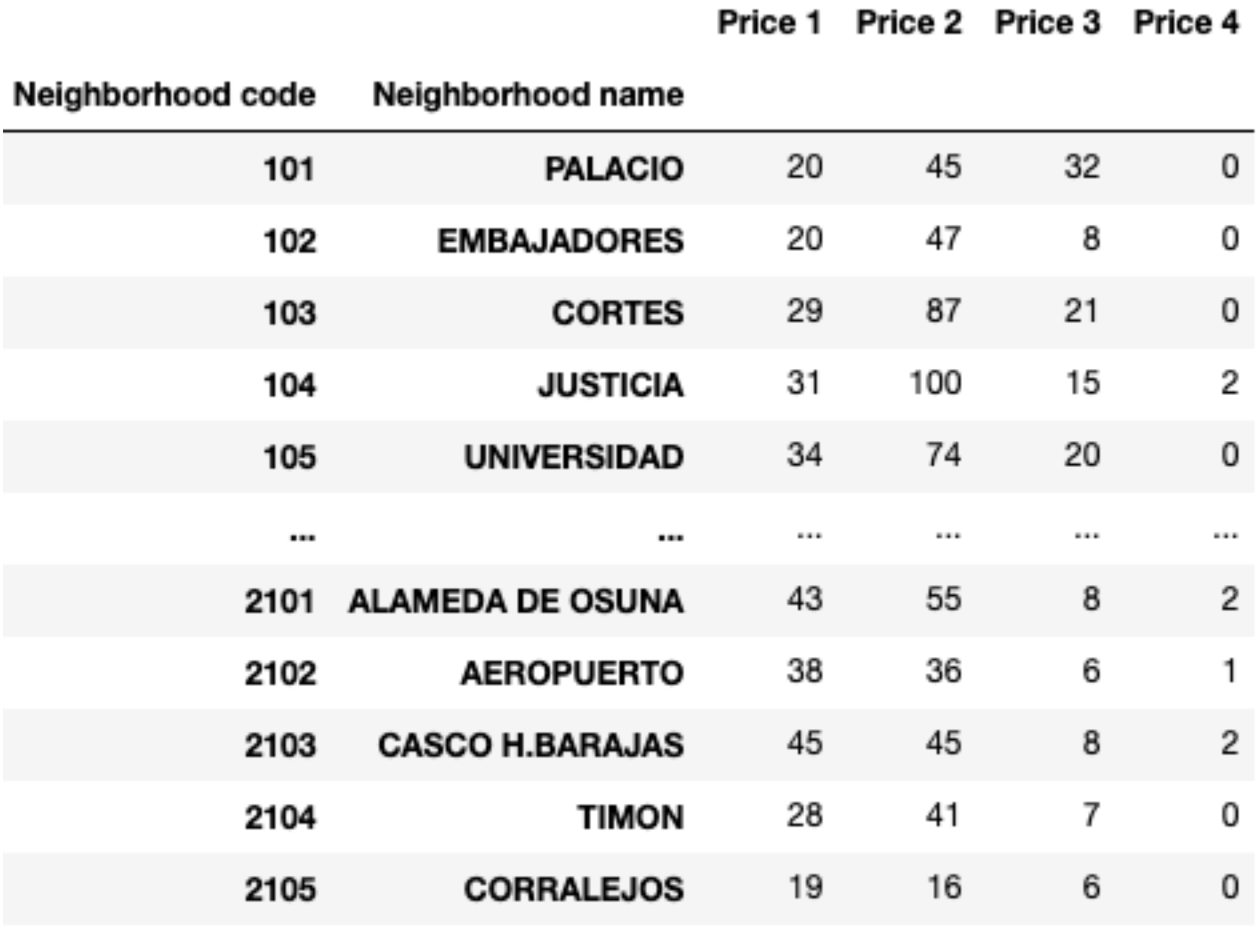
Figure 3. Dataframe of venue categories

We will use this information to classify the neighborhoods into 5 clusters using the K-Means classifier. Before the classification, we normalized the data using the Normalizer preprocessor.

```
from sklearn.preprocessing import Normalizer

nm = Normalizer()

venues_categories_tr = nm.fit_transform(venues_categories_sum)

kmeans = KMeans(n_clusters=5, random_state=0).fit(venues_categories_tr)
```

## Clustering neighborhoods by venue price

In this case, we performed 4 requests per neighborhood, one per each price category. Each request returns all recommended venues with a given price category in the neighborhood. In this case, we are interested in the number of restaurants per neighborhood organized by price. The following image presents a dataframe indexed by neighborhood, where columns are price categories and values are the total number of venues of the given price category in the corresponding neighborhood.



| | | Price 1 | Price 2 | Price 3 | Price 4 |
|---|---|---|---|---|---|
| Neighborhood code | Neighborhood name | | | | |
| 101 | PALACIO | 20 | 45 | 32 | 0 |
| 102 | EMBAJADORES | 20 | 47 | 8 | 0 |
| 103 | CORTES | 29 | 87 | 21 | 0 |
| 104 | JUSTICIA | 31 | 100 | 15 | 2 |
| 105 | UNIVERSIDAD | 34 | 74 | 20 | 0 |
| ... | ... | ... | ... | ... | ... |
| 2101 | ALAMEDA DE OSUNA | 43 | 55 | 8 | 2 |
| 2102 | AEROPUERTO | 38 | 36 | 6 | 1 |
| 2103 | CASCO H.BARAJAS | 45 | 45 | 8 | 2 |
| 2104 | TIMON | 28 | 41 | 7 | 0 |
| 2105 | CORRALEJOS | 19 | 16 | 6 | 0 |

Figure 4. Dataframe of price categories

We used the StandardScaler to standardize the values, and then a K-Means classifier to classify the neighborhoods into 4 groups.

In both cases, we decided to use K-Means as it is a good, cheap solution for pre-clustering and getting some directions that might be explored later with

# 4. Results

## Venue category clusters

Using machine learning algorithms, we have clustered the neighborhoods of Madrid into 5 groups based on the category and number of the restaurants. As we can see in the following image, the clusters are grouped geographically.

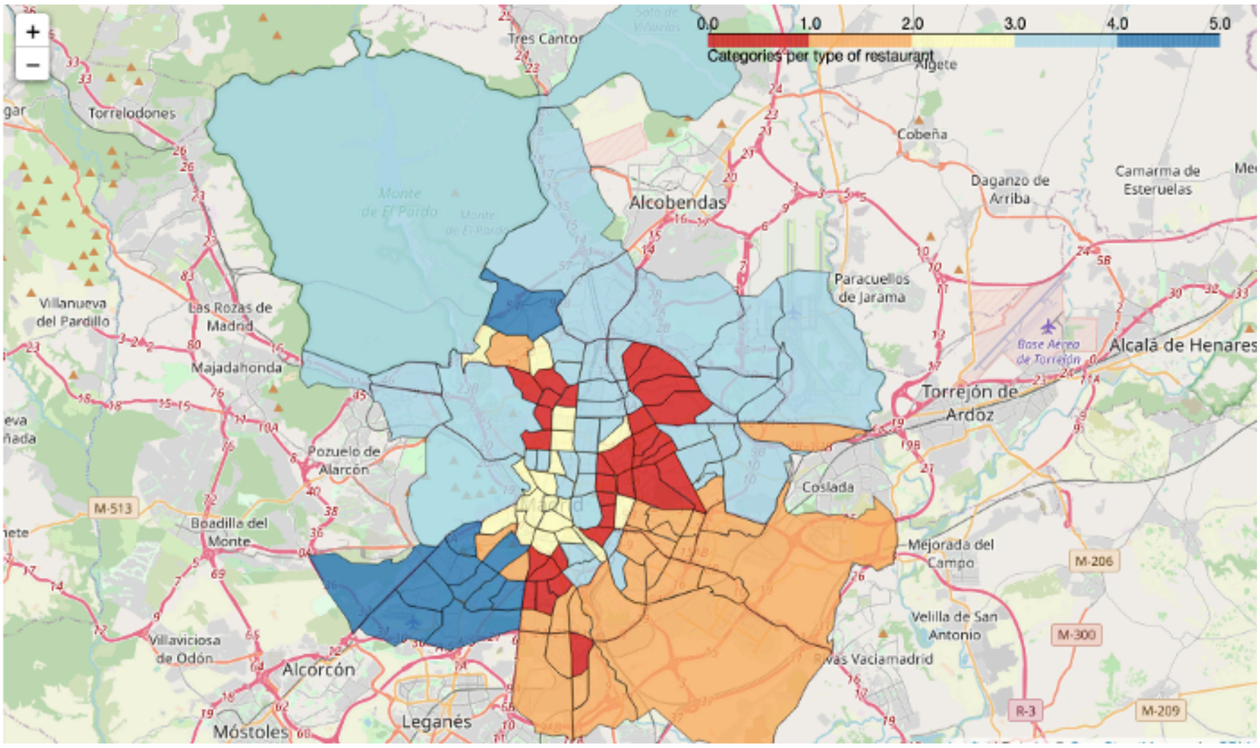


Figure 5. Map of neighborhoods clusters per category

We have also created two heatmaps for visualizing the category of venues in each neighborhood. The first heatmap presents the total number of restaurants per neighborhood per category for the 40 categories most popular.

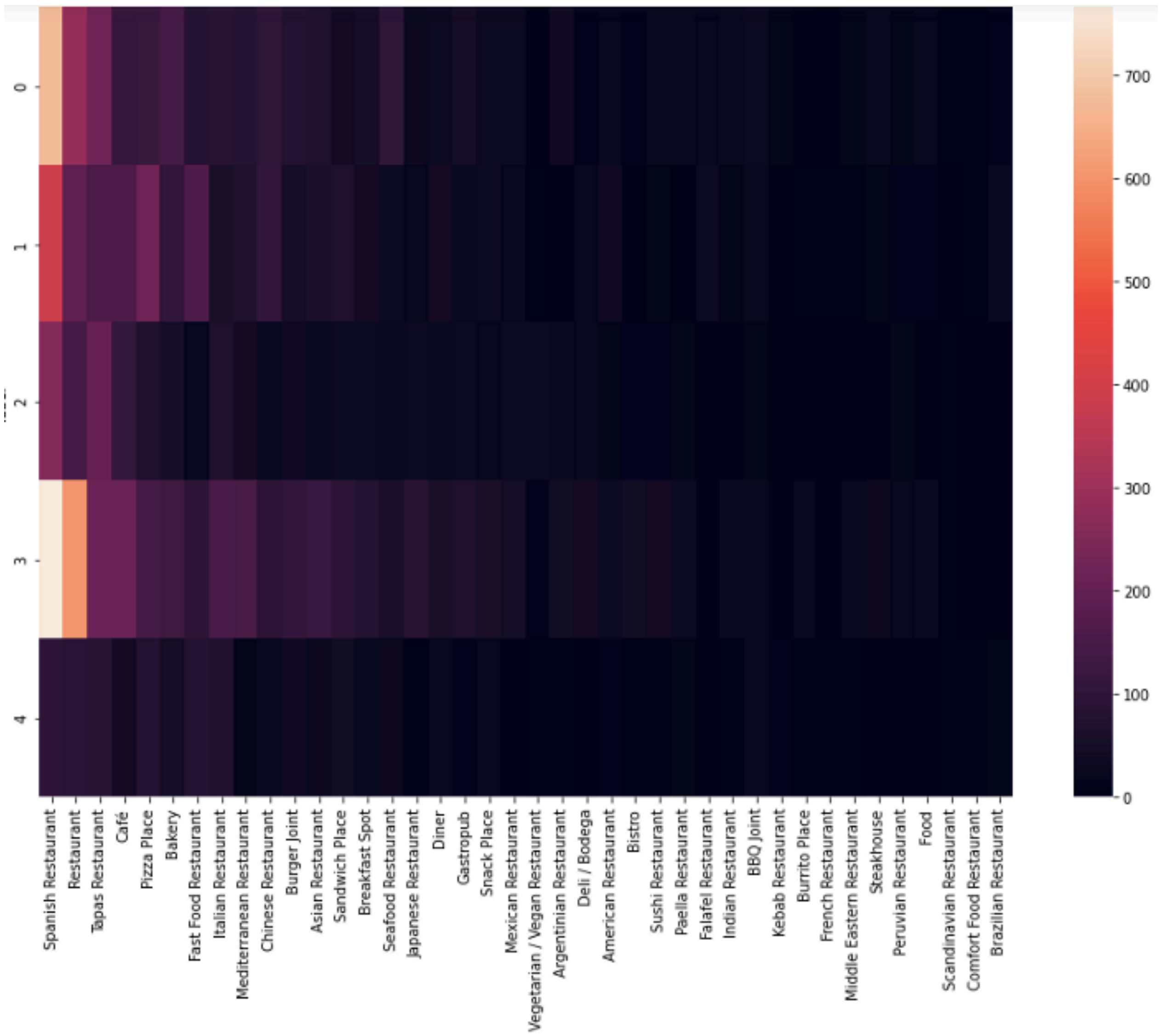


Figure 6. Total number of venues per category per cluster

As the difference between the most popular categories and the less popular categories is quite big, we created a heatmap with the ratio of categories per label:

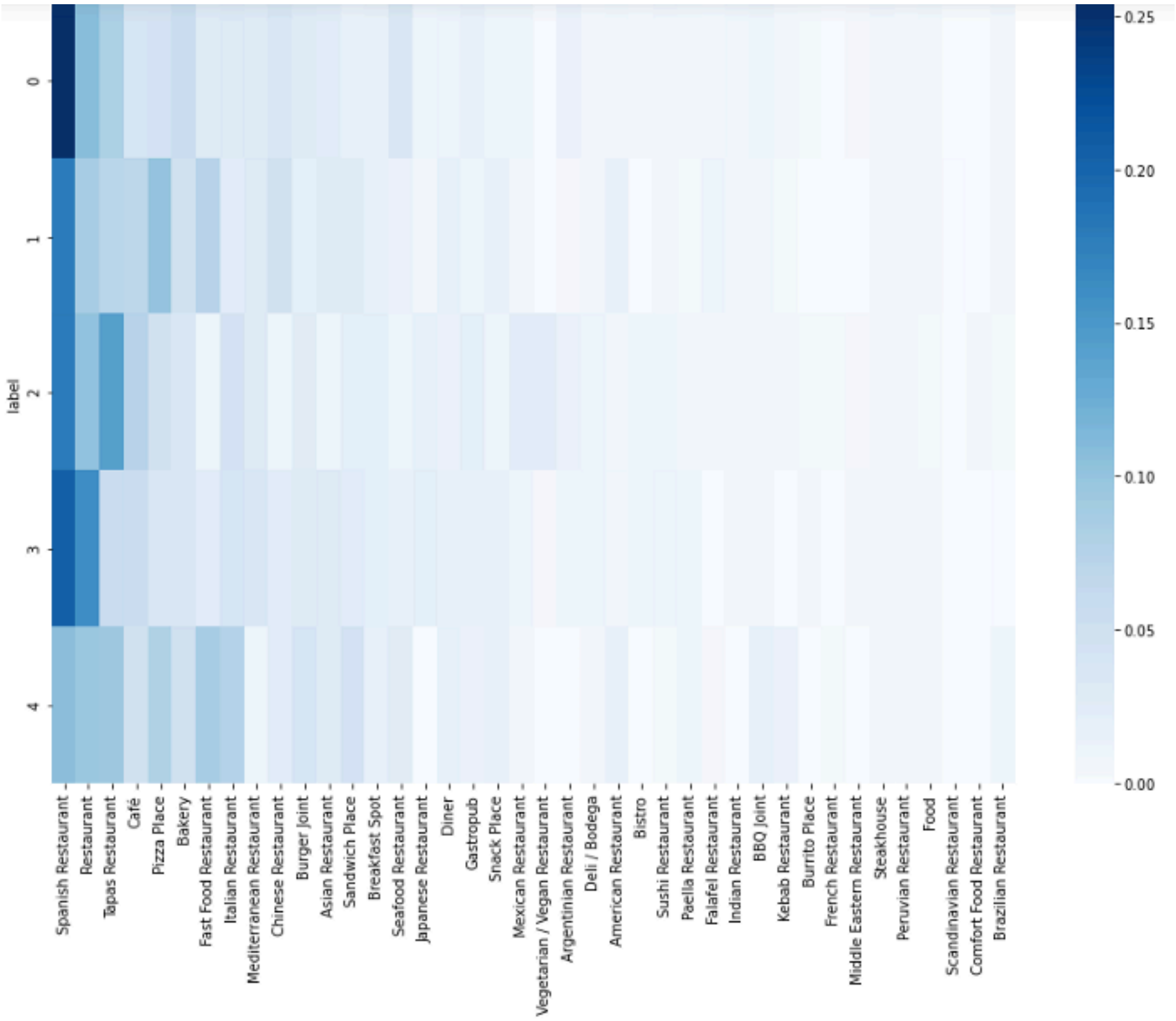


Figure 7. Heatmap with proportional number of categories

- **Cluster 0** comprises **30** neighborhoods, and are in areas surrounding the center of the city; they host the greatest ratio and total number of Spanish restaurants. It also has a higher proportion of breakfast restaurants and bakeries (these are generally working neighborhoods)
- **Cluster 1** comprises **30** neighborhoods, mostly in the south east of the city; they have the highest percentages of pizza places, fast food restaurants and Chinese restaurants.
- **Cluster 2** comprises **18** venues, mostly in the center of the city. They have a great proportion of tapas restaurants (very appreciated by tourists), and the greatest proportion of mexican restaurants.

- **Cluster 3** is the largest one, with **42** neighborhoods, mostly in the north and center of the city. They have a greater proportion, and a great total number, of Spanish restaurants, and a variety of other restaurants.
- **Cluster 4** comprises **11** neighborhoods, mostly in the south west of the city. They have the lowest number of restaurants, but they seem to have more variety of restaurants.

## Venue price clusters

Using the K-Means algorithm, we have clustered the neighborhoods in Madrid based on the number of restaurants of different prices in Madrid. The following image shows the different categories in a map. In this case, the geographic distribution does not seem so clear.

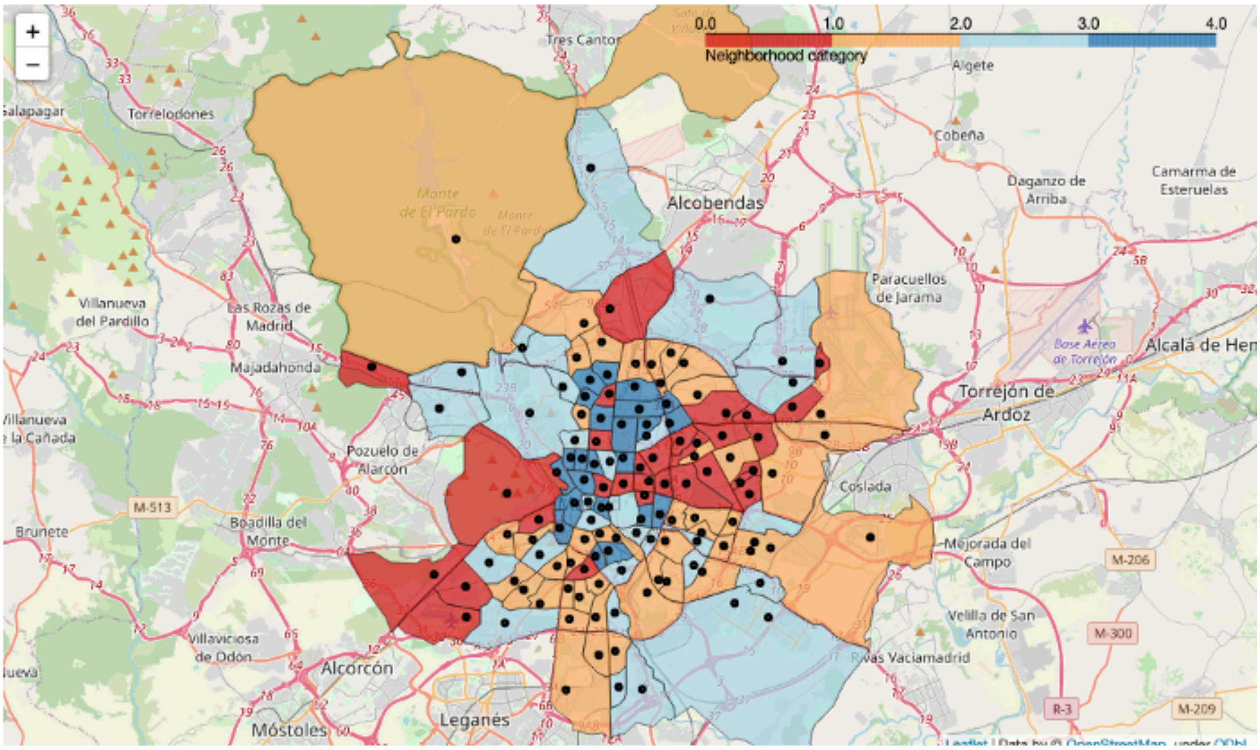


Figure 9. Map of neighborhoods clustered by price category of venues

In this case, to compare the different groups we created a set of box plots that show the distribution of the prices per neighborhood, as shown in the following figure:

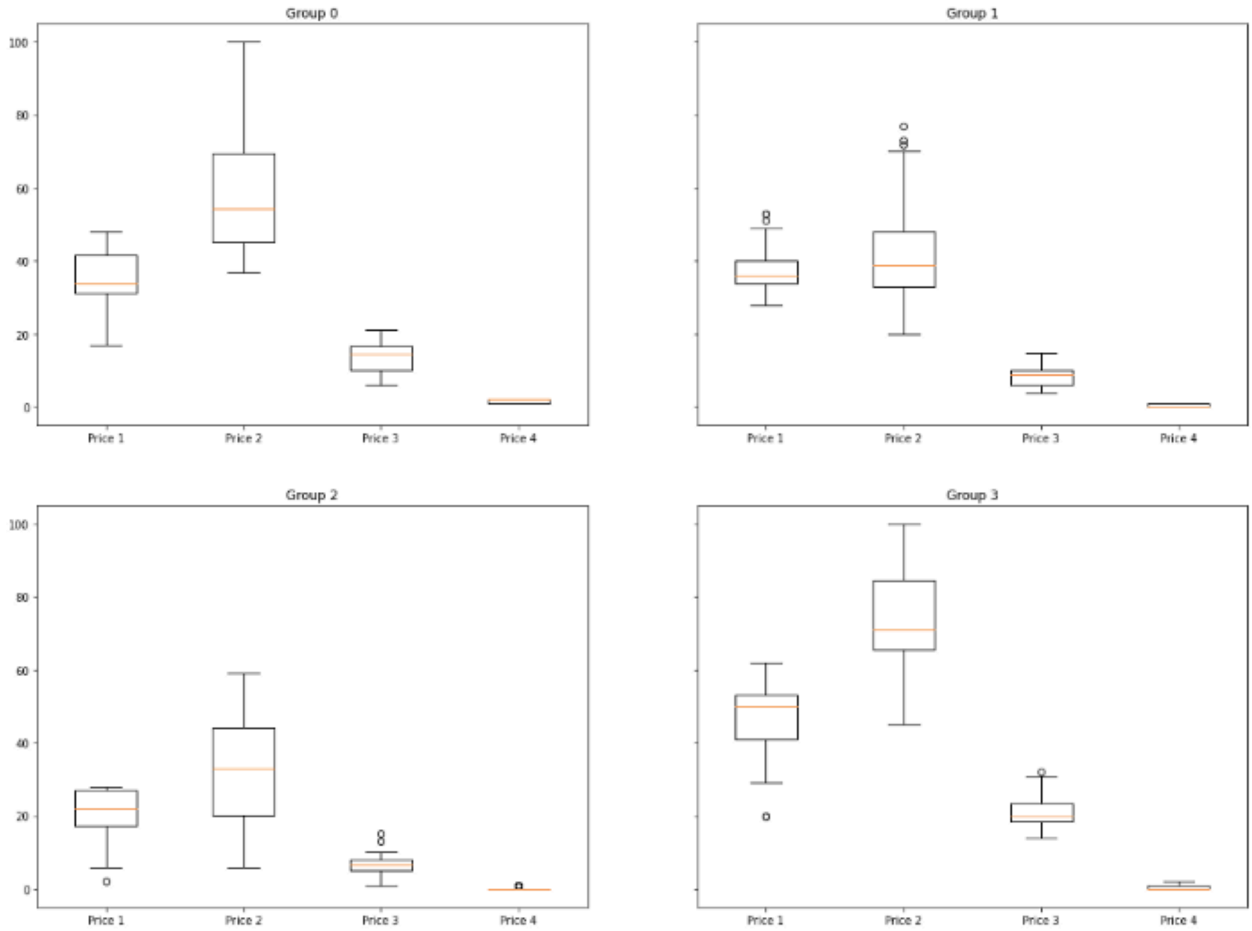


Figure 8. Distribution of prices per cluster

- **Cluster 0** comprises **30** neighborhoods. These neighborhoods have more expensive restaurants (prices 3 and 4).
- **Cluster 1** comprises **48** neighborhoods. They have a similar number of restaurants of price 1 and 2, and a relatively low number of expensive restaurants.
- **Cluster 2** comprises **23** neighborhoods, and have the lowest number of restaurants of all, very few of them in categories 3 and 4.

- **Cluster 3** comprises **23** neighborhoods. They have the highest number of restaurants, with relatively a highest number of restaurants of category 2, and a total high number of restaurants of category 3.

# 5. Discussion

Using K-Means, we can easily cluster different types of data and get a glimpse of the characteristics that can be further explored using more complex clustering algorithms.
In this case, if you know the city the clustering performed makes sense, as the most expensive venues are located in the most touristic and expensive neighborhoods, while working-class neighborhoods have higher proportions of breakfast restaurants and bakeries. Touristic neighborhoods have more 'Tapas' restaurants, while neighborhoods with higher rates of immigration tend to have a higher ratio of non-Spanish restaurants.
For future steps, it would be useful to further analyze the validity of the Foursquare data, looking for duplicates or out-of-date data.

# 6. Conclusions

One of the most challenging aspects when working with data is to get the correct data in the correct format, as well as to clean them. In this case, getting the coordinates of the neighborhoods in Madrid has been complicated, as Nominatim was not as accurate as needed. Another challenging aspect is to find the most adequate visual representation for the data.
All code and data are available here.