The battle of neighborhoods - Barcelona

Eleftheria Exarchou
3 June 2020

Introduction: Business Problem

Barcelona, the capital of Catalonia, has a population of 1.6M people and is at the heart of a metropolitan region of 5M inhabitants.

The cosmopolitan, diverse and intercultural spirit of Barcelona can be seen in the fact that 18.5% of the city's residents are foreign, exceeding 300.000 residents.

Despite the large and ethnically diverse population there is only a handful of Greek restaurants which offer a high quality menu for a middle level target audience.

The stake holder wants to fill this gap by opening a greek restaurant. Criteria to be considered:

- Density of other restaurants
- Other greek, or similar cuisine (e.g. spanish, mediterranean) restaurants in the neighborhood
- Population density
- Distance from city centre

Data

For this project I use data from 2 different sources :

- wikipedia page for districts/neighborhoods in the city
- https://www.bcn.cat/estadistica/, the official page of Barcelona city, for population data

Methodology

We use the package *Beautiful Soup* for web scraping tables with neighborhood/district data. To find the corresponding longitude/latitudes of these data we use the *Nominatim geolocator module*. For visualization, we use the *Folium* package. The machine learning technique used is the *k-means* clustering from *sklearn* package. The rest of analysis relies on pandas, numpy, matplotlib packages.

Analysis

The first step is to create a dataframe with all the neighborhoods and districts in Barcelona.
I do this by scraping the names from the wikipedia page, and using the Nominatim module to find the corresponding latitudes/longitude (Fig 1).

```
# extract data using Beautiful Soup
url='https://en.wikipedia.org/wiki/Districts_of_Barcelona'
res = requests.get(url)

soup = BeautifulSoup(res.content,'lxml')
table = soup.find_all('table')
data = pd.read_html(str(table))
df = pd.DataFrame(data[7])
```

Fig 1: Extracting data from wikipedia using the Beautiful Soup package

For obtaining the longitudes/latitudes of those neighborhoods we use the Nominatim geolocator (Fig 2) and construct the resulting dataframe (Fig 3). We visualize the resulting neighborhoods with the Folium package (Fig 4).

In order to assess the information for the restaurants in the city we use Foursquare, and the resulting dataframe with the restaurants their longitude/latitude (obtained with Nominatim) and the distance from city centre in km is shown in Fig 5.

```
# address = 'Sant Andreu de Palomar,'
def find_lon_lat(address):
    geolocator = Nominatim(user_agent="ny_explorer ")
    location = geolocator.geocode(address,timeout=10000)
    latitude = location.latitude
    longitude = location.longitude
    return [latitude, longitude]
# test it
find_lon_lat('El Coll Barcelona, Spain')
```

Fig 2: Using Nominatim geolocator to identify the latitudes/longitudes of the neighborhoods in Barcelona

```
new_df.tail ()
```

	Districts	Neighborhoods	Latitude	Longitude
77	Sant Martí	El Poblenou	41.400527	2.201729
78	Sant Martí	Provençals del Poblenou	41.412360	2.204885
79	Sant Martí	Sant Martí de Provençals	41.416519	2.198968
80	Sant Martí	La Verneda i la Pau	41.423220	2.202940
81	Sant Martí	La Vila Olímpica del Poblenou	41.389868	2.196846

Fig 3: The resulting data frame for the neighborhoods and districts in Barcelona along with longitudes and latitudes

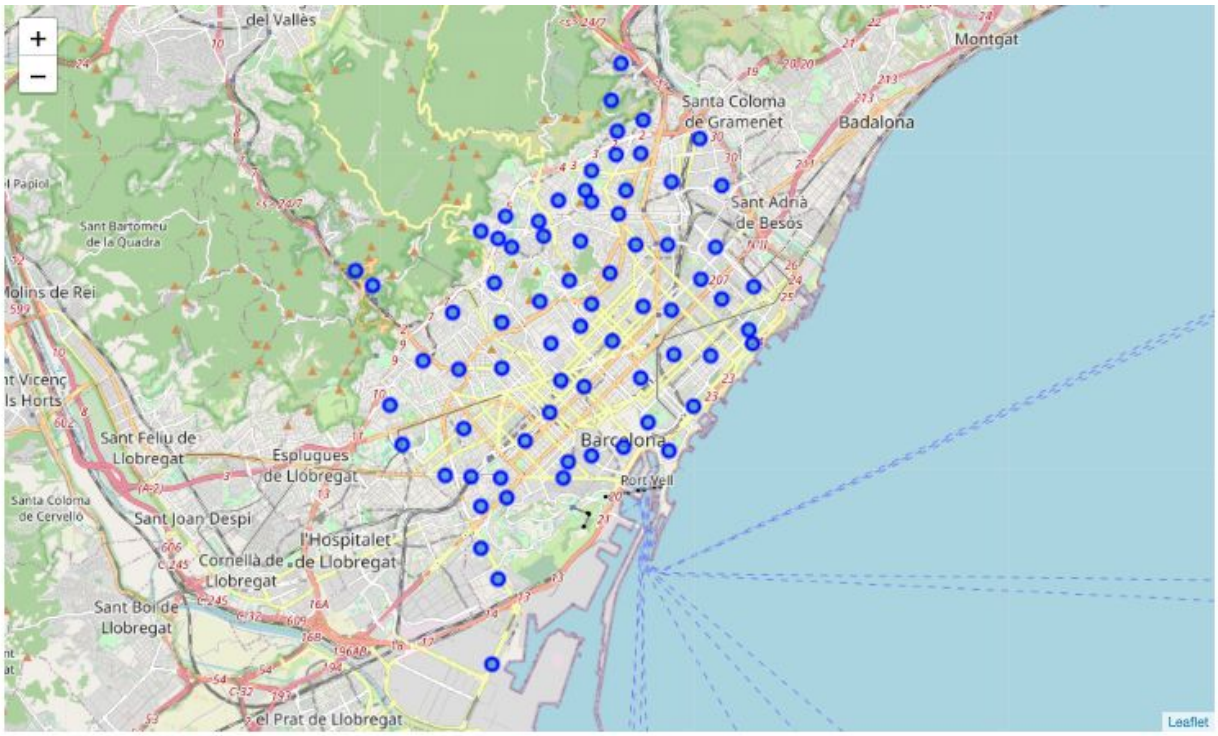

Fig 4: The neighborhoods visualized with the use of the Folium package

```
# Append distance to bcn_restaursnts
bcn_restaurants.insert(3, 'Distance from centre', distance)
bcn_restaurants.head()
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Distance from centre	Venue	Venue Latitude	Venue Longitude	Venue Category
2	La Barceloneta	41.380653	2.189927	1.8	Somorrostro	41.379156	2.189100	Spanish Restaurant
3	La Barceloneta	41.380653	2.189927	1.8	La Cova Fumada	41.379254	2.189254	Tapas Restaurant
5	La Barceloneta	41.380653	2.189927	1.8	Rumbanroll	41.380597	2.187807	Mediterranean Restaurant
7	La Barceloneta	41.380653	2.189927	1.8	La Bombeta	41.380521	2.187573	Tapas Restaurant
8	La Barceloneta	41.380653	2.189927	1.8	La Barra Carles Abellan	41.379838	2.187712	Restaurant

Fig 5: Dataframe with the restaurants of Barcelona (obtained from Foursquare), their longitude/latitude (obtained with Nominatim) and the distance from city centre in km.

A criteria we choose for the choice of the neighborhood is the distance from the city centre. We decide to drop off neighborhoods further than 4 km from the city centre.

Analyze Neighborhoods

We apply one hot encoding (Fig 6); To the resulting dataframe we group rows by neighborhood and by taking the mean of the frequency of occurrence of each category. We apply k-means clustering (Fig 7), and visualize the result (Fig 8).

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

# add neighborhood column back to dataframe
bcn_onehot['Neighborhood'] = bcn_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [bcn_onehot.columns[-1]] + list(bcn_onehot.columns[:-1])
bcn_onehot = bcn_onehot[fixed_columns]


print (bcn_onehot.shape)
```

```
(744, 60)
```

Fig 6: One hot encoding for applying the k-means clustering.

```
# import k-means from clustering stage
from sklearn.cluster import KMeans
```

```
# set number of clusters
kclusters = 7

bcn_grouped_clustering = bcn_grouped.drop('Neighborhood', 1)

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

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

```
array([2, 0, 2, 6, 6, 6, 5, 6, 1, 2], dtype=int32)
```

Fig 7: Applying the k-means clustering.

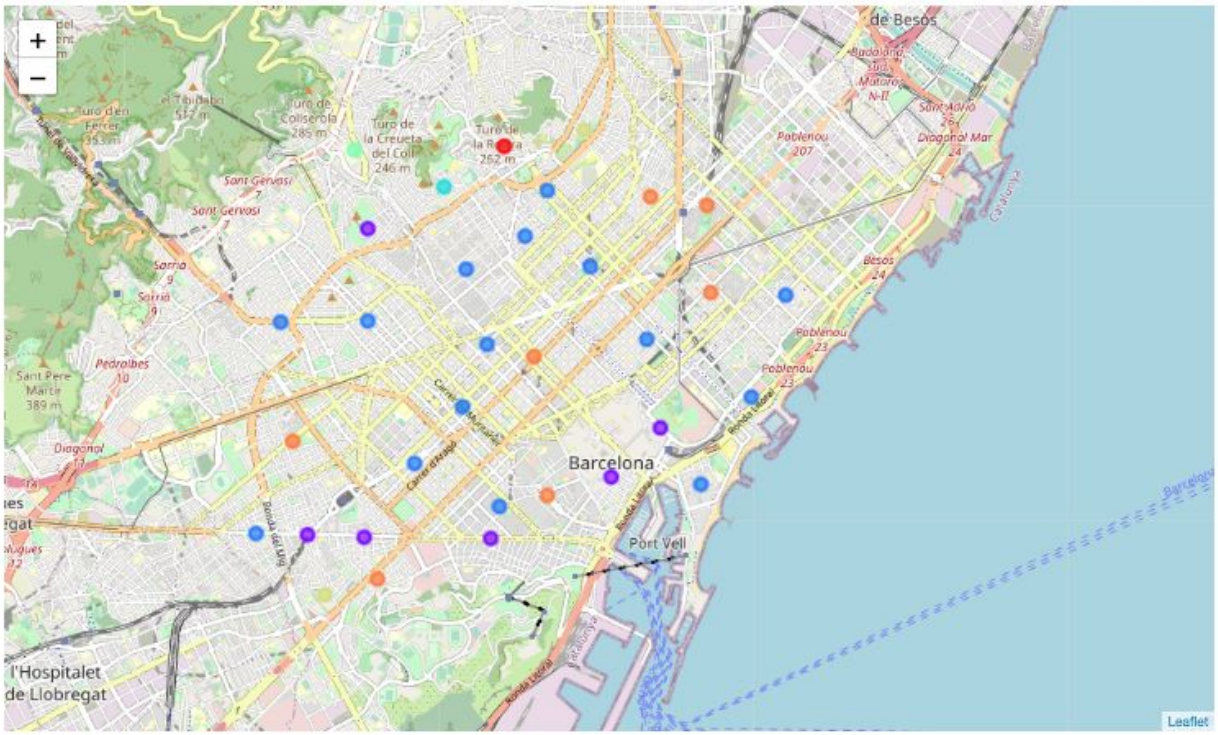

Fig 8: Visualizing the clusters resulting from the k-means algorithm.

Analysis of the Clusters

We want to see the frequency of venues similar to Greek (italian, tapas, spanish, mediterranean) at the different clusters (excluding the 7th cluster, which is empty). Density here is Greek-similar restaurants of each neighborhood in the cluster divided by the total number of restaurants in the same neighborhood. An example looks like Fig 9.

```
cluster1= bcn_merged.loc[bcn_merged['Cluster Labels'] == 1]

t1 = cluster1[ cluster1['Venue Category'].str.contains("greek|italian|tapas|mediterranean|spanis
h", case=False)].\
groupby('Neighborhood').count ()

t2 = cluster1.groupby('Neighborhood').count()

# t1/t2 is the ratio of similar-to-greek cuisine restaurants to total restaurants in each neighbo
rhood
t1/t2
# cluster1[ cluster1['Venue Category'].str.contains("greek|italian|tapas|mediterranean|spanish",
case=False)].\
# groupby('Neighborhood').count ()/test
```

	Neighborhood Latitude_x	Neighborhood Longitude_x	Distance from centre_x	Venue	Venue Latitude_x	Venue Longitude_x	Venue Category	Cluster Labels	Neighborl Latitude_
Neighborhood									
El Poble-sec	0.695652	0.695652	0.695652	0.695652	0.695652	0.695652	0.695652	0.695652	0.695652
Gothic	0.818182	0.818182	0.818182	0.818182	0.818182	0.818182	0.818182	0.818182	0.818182
Hostafrancs	0.565217	0.565217	0.565217	0.565217	0.565217	0.565217	0.565217	0.565217	0.565217
Santa Caterina i la Ribera	0.625000	0.625000	0.625000	0.625000	0.625000	0.625000	0.625000	0.625000	0.625000
Sants	0.550000	0.550000	0.550000	0.550000	0.550000	0.550000	0.550000	0.550000	0.550000
El Putget i Farró	0.615385	0.615385	0.615385	0.615385	0.615385	0.615385	0.615385	0.615385	0.615385

Fig 9: Density of similar-to-greek restaurants per neighborhood for cluster 1.

We exclude clusters other than 2&6 (and the empty 7) because they have high density of similar-to-greek venues. What about Cluster 3?

It seems like La Salut (only neighborhood in Cluster 3), with its only 5 restaurants, is a residential area, so we also exclude it. We are left with cluster 2 & 6. Cluster 6 seems to have a higher density of similar-to-greek venues per neighborhood compared to cluster 2, so we leave it out of the analysis and keep cluster 2.

Results and discussion

We further refine the search by using the additional criteria:

1. The population in each neighborhood/cluster
2. The total number of restaurants in each neighborhood/cluster

Ideally we should opt for the cluster where restaurants/population is small, thus more potential customers.

Let's take population data from Barrios (73) I downloaded them into an csv file, cleaned it and saved locally (Fig 10).

```
pop = pd.read_csv("Population_barrios_Bcn.txt", sep=',\t+',delimiter=',')
# pop[['Neighborhood']]

pop['Neighborhood'] = pop["Neighborhood"].str.strip()
pop['Population'] = pop["Population"].str.strip()
pop
```

	Neighborhood	Population
0	el Raval	48.297
1	el Barri Gòtic	19.180
2	la Barceloneta	15.173
3	Sant Pere Santa Caterina i la Ribera	23.170
4	el Fort Pienc	32.649
...	...	...
68	Diagonal Mar i el Front Marítim del Poblenou	13.625
69	el Besòs i el Maresme	24.660
70	Provençals del Poblenou	21.303
71	Sant Martí de Provençals	26.168
72	la Verneda i la Pau	28.883

73 rows × 2 columns

Fig10: Dataframe with population per neighborhood.

We examine the density (venues/population) per neighborhood (Fig 11) and find that the Neighborhoods with less density (venues/population) is number 2, 4 & 3 (Fig 12).

```
testdf = cluster2_pop.groupby('Population').count().reset_index()
testdf1= testdf['Population'].astype(float)
testdf2= testdf['Neighborhood']
testdf2/testdf1
# cluster2_pop.head()
```

```
0    3.361234
1    0.808003
2    0.312509
3    0.700099
4    0.610200
dtype: float64
```

Fig 11: Venues per population for the 5 neighborhoods in the chosen cluster 2.

```
pop_reduced[pop_reduced['Population'].astype(float)==testdf1[2]]
```

	Neighborhood	Population
31	Camp d'en Grassot i Gràcia Nova	35.199

```
pop_reduced[pop_reduced['Population'].astype(float)==testdf1[4]]
```

	Neighborhood	Population
30	Vila de Gràcia	50.803

```
pop_reduced[pop_reduced['Population'].astype(float)==testdf1[3]]
```

	Neighborhood	Population
9	Sant Antoni	38.566

Fig 12: the neighborhoods with the less density of venues/population.

Conclusion

After the final refinement, based on the number of restaurants per population, our final conclusion consists of 3 neighborhoods:

- **Camp d'en Grassot i Gràcia Nova**
- **Vila de Gràcia**
- **Sant Antoni**

Of course in this analysis we did not take other factors into account, like: tourist movement, number of hotels in the area, rental/buying prices for property, public transport and accessibility, crime rate. Based on the machine learning techniques, the conclusion above can be a first step to a more thorough analysis.