The battle of the neighborhoods

Finding the best location to open a restaurant in Madrid, Spain.

FEBRUARY 3

IBM COURSERA DATA SCIENCE CAPSTONE

Authored by: Luis Castillo

IBM

coursera

Contents

1. BUSINESS PROBLEM

Our customer, owner of the Italian restaurant "XYZ Fancy Dining" is interested in opening a new venue in Madrid. Madrid is one of the busiest cities in Europe, with more than three million residents and an average of almost 800.000 visitors each month.

This would be our customer's second restaurant location, after having successfully opened a venue in El Carme, a very lively neighborhood from Valencia city.

Considering that our customer has had very good results with their Valencia location, they have requested our data science team to find a neighborhood with similar characteristics.

The problem question would be: **What neighborhood from Madrid has the most similar characteristics in terms of entertainment and dining options compared to El Carme in Valencia City?**

2. DATA

The data to be used for this project comes from three different locations:

- Foursquare. It is a local search-and-discovery service which provides information on different types of entertainment, drinking and dining venues. Foursquare has an API that can be used to query their database and find information related to the venues, such as location, overall category, reviews and tips.
- Madrid Neighborhood Names and geographic coordinates. Available on https://datos.madrid.es/, this is used to obtain the neighborhood location information from the city.
- Valencia City Neighborhood Names and geographic coordinates. Data available on http://mapas.valencia.es/lanzadera/opendata/Barrios/SHAPE
- Madrid census data, were we can get the population and income statistics, available in http://www-2.munimadrid.es/CSE6/jsps/menuBancoDatos.jsp

Below the details of how we will use each data source during this project.

2.1. Foursquare API data

For this project we will use the Foursquare Places API. One of the features of this API is to provide a list of venues within a specific location, based on the Lat/Lon coordinates and a radius.

In order to obtain a list of venues within a specified area, we use the "explore" endpoint from the API. By passing the proper parameters via an HTTP request to the *explore* endpoint, we get a JSON object with the information shown in the table below:

Field	Description
id	A unique string identifier for this venue.
name	The best known name for this venue.
location	An object containing none, some, or all of `address` (street address), `crossStreet`, `city`, `state`, `postalCode`, `country`, `lat`, `lng`, and `distance`. All fields are strings, except for `lat`, `lng`, and `distance`. Distance is measured in meters. Some venues have their locations intentionally hidden for privacy reasons (such as private residences). If this is the case, the parameter `isFuzzed` will be set to true, and the `lat`/`lng` parameters will have reduced precision.
categories	An array, possibly empty, of categories that have been applied to this venue. One of the categories will have a `primary` field indicating that it is the primary category for the venue. For the complete category tree, see categories.

Figure 1. Information contained in response to request towards "explore" endpoint

The *location* object contains the coordinates of each venue, which will be used to associate it with its respective neighborhood.
The *categories* array will be used to categorize the neighborhood. Basically, we will count how many venues from all available categories are found on each neighborhood, and then use that information to compare neighborhoods from Madrid with El Carme in Valencia.

2.2. Madrid Neighborhoods

The Madrid city government has made available to the public a series of datasets with information of interest. We will be using the “Divisiones administrativas: distritos, barrios y divisiones históricas” dataset, available in the following URL: https://datos.madrid.es/egob/catalogo/200078-10-distritos-barrios.zip.
The data insinde the .zip file is in ESRI format. To convert this to a dataframe that we can use, *geopandas* python library.

2.3. Valencia City Neighborhoods

Valencia City Neighborhood Names and geographic coordinates. Data available on
http://mapas.valencia.es/lanzadera/opendata/Barrios/SHAPE.
This data is also available in ESRI format.

2.4. Madrid Census data

To complement our analysis we will be using the statistics of the population and average income per neighborhood in madrid. This data is available in the municipality data bank,
http://www-2.munimadrid.es/CSE6/jsps/menuBancoDatos.jsp

3. METHODOLOGY

3.1 Data preprocessing

- ### Neighborhood basic information and census data

During the data preprocessing stage, we prepare the data to be used during the machine learning process. The data structure for the neighborhood information is different between Madrid and Valencia, so we need to adapt both of them.

```
#Import Neighborhoods geodata
madrid_neighborhoods = gpd.read_file("data/neighborhoods-madrid/BARRIOS.shp")
```

Lets get some basic information on the imported data

```
madrid_neighborhoods.head(3)
```

	OBJECTID	geodb_oid	CODDIS	NOMDIS	CODBAR	CODDISTRIT	CODBARRIO	NOMBRE	ORIG_FID	geometry
0	108	108	17	Villaverde	172	17	17-2	San Cristobal	107	POLYGON ((441930.8668000005 4466853.1887, 4419...
1	109	109	17	Villaverde	173	17	17-3	Butarque	108	POLYGON ((444144.8566044134 4464473.210504748,...
2	111	111	17	Villaverde	175	17	17-5	Los Angeles	110	POLYGON ((441147.7280000008 4466374.483400001,...

Figure 2. Madrid Neighborhood Data

```
# Import neighborhoods
val_neighborhoods = gpd.read_file("http://mapas.valencia.es/lanzadera/opendata/Barrios/SHAPE")
```

Lets get some basic information on the imported data

```
val_neighborhoods.head(3)
```

	codbarrio	nombre	coddistbar	coddistrit	geometry
0	1	BENIFARAIG	171	17	POLYGON ((725499.03 4378693.39, 725477.797 437...
1	1	BENICALAP	161	16	POLYGON ((725164.733 4375392.58, 725187.044 43...
2	2	TORREFIEL	152	15	POLYGON ((726040.348 4375385.446, 725995.041 4...

Figure 3. Valencia Neighborhood Data

After preprocessing, we came with the following dataset, containing the needed information for both cities:

```
#Concatenate two dataframes
neighborhoods = pd.concat([madrid_neighborhoods,val_neighborhoods])
```

```
#Check random neighborhoods Madrid
neighborhoods[neighborhoods['City']=='Madrid'].head(3)
```

	District	Neighborhood	Longitude	Latitude	City
0	VILLAVERDE	SAN CRISTOBAL	-3.688372	40.340888	Madrid
1	VILLAVERDE	BUTARQUE	-3.676254	40.337115	Madrid
2	VILLAVERDE	LOS ANGELES	-3.699137	40.355790	Madrid

```
#Check random neighborhoods Valencia
neighborhoods[neighborhoods['City']=='Valencia'].head(3)
```

	District	Neighborhood	Longitude	Latitude	City
0	17	BENIFARAIG	-0.384621	39.525644	Valencia
1	16	BENICALAP	-0.391002	39.493006	Valencia
2	15	TORREFIEL	-0.376932	39.495198	Valencia

Figure 4. "Neighborhoods" dataset, containing both Madrid and Valencia

As shown in Figure 4, during the data pre-processing stage we created a dataset containing each of the neighborhood's basic information for both cities.

Finally, we create a dataset that contains the census data per neighborhood, as shown below.

```
madrid_income = pd.read_excel('data/income-madrid/income-madrid.xls')
madrid_income.head()
```

	District	Neighborhood	Average Income
0	CENTRO	PALACIO	34675.85
1	CENTRO	EMBAJADORES	25999.83
2	CENTRO	CORTES	34952.68
3	CENTRO	JUSTICIA	40314.88
4	CENTRO	UNIVERSIDAD	30701.65

Figure 5. Income per neighborhood – Madrid

z.munimadrid.es/CSEU/jsps/menuBancoDatos.jsp

```
madrid_population = pd.read_excel('data/population-madrid/population-madrid.xls', skipfooter=4, skiprows=4)
madrid_population.head()
```

	Distrito	Barrio	Edad	Total
0	CENTRO	PALACIO	Total	22984
1	CENTRO	EMBAJADORES	Total	45433
2	CENTRO	CORTES	Total	10525
3	CENTRO	JUSTICIA	Total	17205
4	CENTRO	UNIVERSIDAD	Total	31809

Figure 6. Population per neighborhood - Madrid

- **Foursquare API data**

By using a custom function, calling the "explore" endpoint, we created a dataset with the top 100 venues within 500 meters of the center of each neighborhood.

	Neighborhood	District	City	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	SAN CRISTOBAL	VILLAVERDE	Madrid	40.340888	-3.688372	Cercanías San Cristóbal de Los Ángeles	40.341710	-3.683878	Train Station
1	SAN CRISTOBAL	VILLAVERDE	Madrid	40.340888	-3.688372	Igreen Aire Acondicionado y Climatización	40.341581	-3.686213	Furniture / Home Store
2	SAN CRISTOBAL	VILLAVERDE	Madrid	40.340888	-3.688372	Bar Vietnam	40.341090	-3.686568	Snack Place
3	SAN CRISTOBAL	VILLAVERDE	Madrid	40.340888	-3.688372	El Rincón de Peri	40.342427	-3.691998	Breakfast Spot
4	BUTARQUE	VILLAVERDE	Madrid	40.337115	-3.676254	Mercadona	40.340165	-3.675179	Grocery Store

Figure 7. Sample of the "venues" dataset

By using a custom function, calling the "explore" endpoint, we created a dataset with the top 100 venues within 500 meters of the center of each neighborhood.

We performed one-hot encoding on the "Venue Category" column, creating a new dataset as shown below.

```
venues_grouped.sample(15)
```

	Neighborhood	Accessories Store	African Restaurant	Airport	Airport Service	American Restaurant	Arcade	Arepa Restaurant	Argentinian Restaurant	Art Gallery	...	Used Bookstore	Vegetarian / Vegan Restaurant	Video Game Store	Video Store	Vietnamese Restaurant	Warehouse Store	Wl
155	PEÑA GRANDE	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.2	0.000000	...	0.0	0.00000	0.0	0.0	0.0	0.0	
201	VALLEHERMOSO	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.000000	...	0.0	0.00000	0.0	0.0	0.0	0.0	
153	PAVONES	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.000000	...	0.0	0.00000	0.0	0.0	0.0	0.0	
185	SANTA EUGENIA	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.000000	...	0.0	0.00000	0.0	0.0	0.0	0.0	
75	EL VISO	0.0	0.0	0.0	0.0	0.038462	0.000000	0.0	0.0	0.038462	...	0.0	0.00000	0.0	0.0	0.0	0.0	
146	PACIFICO	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.029851	...	0.0	0.00000	0.0	0.0	0.0	0.0	
87	FUENTELARREINA	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.000000	...	0.0	0.00000	0.0	0.0	0.0	0.0	
168	REJAS	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.000000	...	0.0	0.00000	0.0	0.0	0.0	0.0	
92	HISPANOAMERICA	0.0	0.0	0.0	0.0	0.000000	0.014706	0.0	0.0	0.000000	...	0.0	0.00000	0.0	0.0	0.0	0.0	
51	CIUTAT UNIVERSITARIA	0.0	0.0	0.0	0.0	0.032258	0.000000	0.0	0.0	0.000000	...	0.0	0.00000	0.0	0.0	0.0	0.0	
16	ARGUELLES	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.000000	...	0.0	0.02439	0.0	0.0	0.0	0.0	
101	LA CARRASCA	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.000000	...	0.0	0.00000	0.0	0.0	0.0	0.0	
152	PATRAIX	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.000000	...	0.0	0.00000	0.0	0.0	0.0	0.0	
53	COMILLAS	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.000000	...	0.0	0.00000	0.0	0.0	0.0	0.0	
140	NOU MOLES	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.000000	...	0.0	0.00000	0.0	0.0	0.0	0.0	

Figure 8. Venues_Grouped dataset (after one-hot encoding)

The dataset from the figure above will be used for clustering later.

3.2 Exploratory Data Analysis

First, let's check the number of neighborhoods we are working with on each city.

```
print('The number of neighborhoods in Madrid is: {}'.format(madrid_neighborhoods['Neighborhood'].nunique()))
print('The number of districts in Madrid is: {}'.format(madrid_neighborhoods['District'].nunique()))
```

```
The number of neighborhoods in Madrid is: 131
The number of districts in Madrid is: 21
```

```
print('The number of neighborhoods in Valencia is: {}'.format(val_neighborhoods['Neighborhood'].nunique()))
print('The number of districts in Valencia is: {}'.format(val_neighborhoods['District'].nunique()))
```

```
The number of neighborhoods in Valencia is: 88
The number of districts in Valencia is: 19
```

Figure 9. Number of neighborhoods per city

There are more neighborhoods in Madrid, being a larger city. However, the number is relatively comparable to Valencia's 88 neighborhoods.

Lets now evaluate the venues dataset to compare both cities:

```
#Get how many venues were found
print('A total of {} venues were found in Madrid'.format(venues[venues['City']=='Madrid'].shape[0]))
print('A total of {} venues were found in Valencia'.format(venues[venues['City']=='Valencia'].shape[0]))
```

```
A total of 3517 venues were found in Madrid
A total of 2553 venues were found in Valencia
```

Figure 10. Number of venues found for each city

And let's compare the distribution of types of venues found on each city:

```python
# Count the number of locations per Venue Category in Madrid
venues[venues['City']=='Madrid'].groupby('Venue Category').count()['Neighborhood'].sort_values(ascending=False).head(10)
```

```
Venue Category
Spanish Restaurant    381
Restaurant            193
Bar                   166
Tapas Restaurant      154
Café                  109
Hotel                 100
Coffee Shop            91
Bakery                 84
Pizza Place            74
Italian Restaurant     73
Name: Neighborhood, dtype: int64
```

```python
# Count the number of locations per Venue Category in Valencia
venues[venues['City']=='Valencia'].groupby('Venue Category').count()['Neighborhood'].sort_values(ascending=False).head(10)
```

```
Venue Category
Spanish Restaurant          178
Tapas Restaurant            153
Restaurant                  113
Mediterranean Restaurant    103
Café                         87
Hotel                        85
Grocery Store                83
Italian Restaurant           80
Bakery                       65
Pub                          59
Name: Neighborhood, dtype: int64
```

Figure 11. Number of venues per category Madrid and Valencia

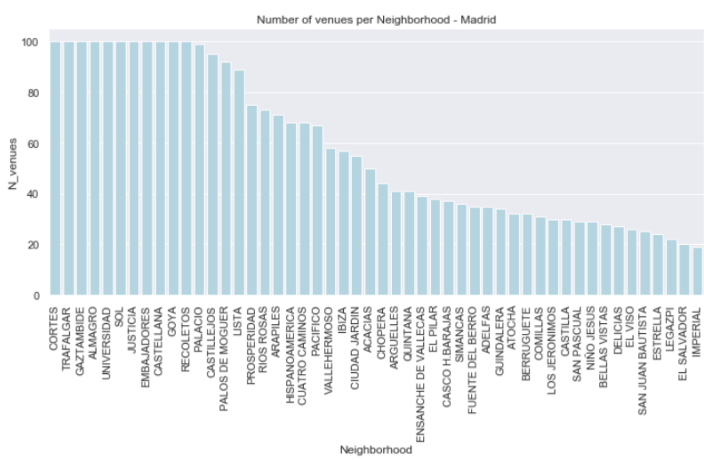
As shown, the types of venues are somehow similar between both cities. Traditional "Spanish restaurant" are in the top for both cities. As shown below, there is slightly more diversity in the types of venues available in Madrid.

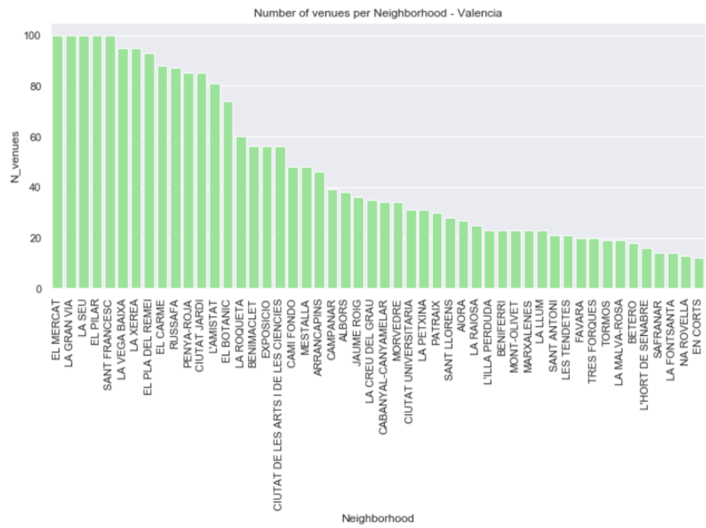
```python
#Number of unique venue categories per city
print('There are {} uniques categories in Madrid.'.format(len(venues[venues['City']=='Madrid']['Venue Category'].unique())))
print('There are {} uniques categories in Valencia.'.format(len(venues[venues['City']=='Valencia']['Venue Category'].unique())))
```

```
There are 269 uniques categories in Madrid.
There are 215 uniques categories in Valencia.
```

Figure 12. Types of venues per city

The graphs below show the top neighborhoods by venues:





3.3 Clustering

With the venues_grouped dataset (one-hot encoding of the venue types per neighborhood) we now group the neighborhoods into clusters using the KMeans Clustering method.
For our project we selected K=20, aiming to desegregate as much as possible the data.

Now lets initialize the k-means model using K=20

```
]: k_means = KMeans(init = "k-means++", n_clusters = 20, n_init = 15)
```

```
]: # Fit the model
   k_means.fit(venues_grouped.drop('Neighborhood',axis=1))
```

```
]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=20, n_init=15, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

```
]: #Add the labels to the venues_grouped dataset
   venues_grouped['Cluster']=k_means.labels_
```

```
]: #Obtain the number of neighborhoods per cluster
   venues_grouped.groupby('Cluster')['Neighborhood'].count()
```

-

Figure 13. Clustering of neighborhood data

The figure below shows the amount of neighborhoods per each cluster. As we can see, there are 5 “dominant” clusters, out of which Cluster 12 has the highest amount of neighborhoods (81). Remember this analysis includes both Madrid and Valencia, now we have to separate only the Madrid results.

```
venues_grouped_count = venues_grouped.groupby('Cluster')['Neighborhood'].count().to_frame()
venues_grouped_count.reset_index(inplace=True)
ax = sns.barplot(x='Cluster', y='Neighborhood', data=venues_grouped_count, color='green')
ax.set_title('Number of neighborhoods per cluster');
```

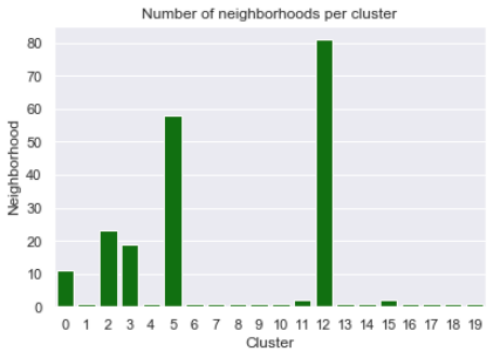


Figure 14. Number of neighborhoods per cluster

We identify that the target neighborhood "El Carme" is located in cluster 12:

```
target_cluster_df = neighborhoods_venues_sorted.loc[neighborhoods_venues_sorted['Neighborhood']=='EL CARME']
target_cluster_df.reset_index(inplace=True)
target_cluster=target_cluster_df.loc[0].at['Cluster']
print('The target cluster is: {} '.format(target_cluster))
```

```
The target cluster is: 12
```

Figure 15. EL Carme cluster location

And finally we can determine the neighborhoods from Madrid that belong to this cluster:

```
#Filter neighborhoods from Madrid that belong to the target cluster
possible_neighborhoods = neighborhoods_venues_sorted[
    (neighborhoods_venues_sorted['Cluster']==target_cluster) &
    (neighborhoods_venues_sorted['City']=='Madrid')]

print('There are {} neighborhoods in Madrid with similar characteristics than El Carme'
      .format(possible_neighborhoods.shape[0]))
```

```
There are 48 neighborhoods in Madrid with similar characteristics than El Carme
```

Figure 16. Total number of potential neighborhoods to open the restaurant in Madrid

The number of neighborhoods for cluster 12 is still high, so we will create a ranking metric to order the list. The ranking will be based on the following criteria:

a) Total Population. Weight: 50%
b) Average income per household within each neighborhood. Weight: 25%
c) Amount of already existing Italian restaurants. Weight: 25%

The first step is to normalize each of the metrics, so they can all be represented with a number from 0 to 1. For (a) and (b) we divide by the maximum value of the total population and income dataset. For (c), we create an index with the rate of non Italian restaurants.

	District	Neighborhood	Population	Population_Normalized
0	CENTRO	PALACIO	22984	0.345406
1	CENTRO	EMBAJADORES	45433	0.682772
2	CENTRO	CORTES	10525	0.158171

	District	Neighborhood	Average Income	Income_Normalized
0	CENTRO	PALACIO	34675.85	0.308722
1	CENTRO	EMBAJADORES	25999.83	0.231478
2	CENTRO	CORTES	34952.68	0.311186

	Neighborhood	Number_Italian_Restaurants	Non_Italian_Restaurants
0	ABRANTES	0	1.000000
1	ACACIAS	1	0.888889
2	ADELFAS	0	1.000000

Figure 17. Normalized metrics used to calculate the ranking

The final step is to combine the three metrics for each of the possible neighborhoods and to order the neighborhoods based on this rank. This is shown in the figure below

```
recommended_neighborhoods
```

	District	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	Population	Population_Normalized	Income_Normalized	Non_Italian_Restaurants	Ranking
0	LATINA	LAS AGUILAS	Spanish Restaurant	Coffee Shop	Sports Club	51703	0.776998	0.266295	1.000000	0.581702
1	SALAMANCA	GUINDALERA	Spanish Restaurant	Bakery	Restaurant	41751	0.627438	0.412576	1.000000	0.558121
2	CHAMARTÍN	EL VISO	Spanish Restaurant	Restaurant	Diner	17325	0.260362	0.922119	1.000000	0.552923
3	CHAMARTÍN	NUEVA ESPAÑA	Tapas Restaurant	Restaurant	Donut Shop	24699	0.371179	0.714457	1.000000	0.535650
4	CHAMARTÍN	HISPANOAMERICA	Spanish Restaurant	Restaurant	Bar	31815	0.478119	0.557647	1.000000	0.534236
5	CARABANCHEL	VISTA ALEGRE	Pizza Place	Cosmetics Shop	Comedy Club	46122	0.693126	0.247401	1.000000	0.533153
6	CENTRO	EMBAJADORES	Bar	Café	Tapas Restaurant	45433	0.682772	0.231478	1.000000	0.522403
7	ARGANZUELA	ACACIAS	Spanish Restaurant	Tapas Restaurant	Bar	36907	0.554642	0.397693	0.888889	0.505403
8	CHAMARTÍN	PROSPERIDAD	Bar	Spanish Restaurant	Tapas Restaurant	36730	0.551982	0.389078	0.888889	0.501057
9	SALAMANCA	CASTELLANA	Spanish Restaurant	Restaurant	Boutique	17161	0.257897	0.737274	1.000000	0.486995

Figure 18. Top 10 neighborhood recommendations to open new italian restaurant in Madrid

4. RESULTS DISCUSSION

After clustering the Madrid and Valencia neighborhoods based on the results from the Foursquare API data, we were able to separate our dataset into 5 distinct clusters, and then from our target cluster pick the best candidates for our customer to open their new Italian restaurant.

As shown in **Figure 18. Top 10 neighborhood recommendations to open new italian restaurant in Madrid** The selected neighborhoods have similar characteristics. Most of them are dominated by Spanish Restaurants and Bars, are densely populated neighborhoods and have few or no Italian restaurants. These constitute good candidates for opening a restaurant.

One issue I noted during the clustering analysis was that, even though we set the KMeans Clustering method with K=20 (aiming to segregate the neighborhoods as much as possible) we found several "one neighborhood" clusters. This denotes the importance of properly selecting the K value when using this method. In our case, it wasn't relevant to redo the analysis with a different K, since we were only looking for one specific cluster.

The ranking procedure shown at the end of section 3.3 was necessary due to the high amount of results found for the target cluster. In our example, we integrated three different metrics (Population, Income and existing Competition). For a real-life project, probably additional metrics could be added to create a more robust index.

5. CONCLUSSION

We were able to determine a good set of ten options to propose to our customer to open a new restaurant, considering the variables described in the previous sections.

During this project I applied several methodologies used during the course, such as data wrangling with pandas, basic data visualization and machine learning techniques.

For future projects with similar characteristics, it should be considered to expand the amount of data available (for example, using the premium features of the Foursquare API) and other clustering algorithms such as DBSCAN.