Capstone Project - The Battle of Neighborhoods

Finding the Best Place to Open A Restaurant in Milan

KHALID ALRIFAI

Table of Contents

1. **INSTRUCTION** 2
2. **INTRODUCCTION** 2
3. **BUSINESS PROBLEM** 3
   - **3.1. Target audience:** 4
   - **3.2. Tools:** 5
   - **3.3. Data Section:** 5
4. **METHODOLOGY** 8
   - **4.1. Business Understanding** 8
   - **4.2. Analytical Approach** 8
   - **4.3. Data Exploration** 8
   - **4.4. Clustering:** 12
5. **RESULTS** 14
6. **CONCLUSION** 16
7. **NEXT DEVELOPMENTS** 16

1. INSTRUCTION

Now that you have been equipped with the skills and the tools to use location data to explore a geographical location, over the course of two weeks, you will have the opportunity to be as creative as you want and come up with an idea to leverage the Foursquare location data to explore or compare neighborhoods or cities of your choice or to come up with a problem that you can use the **Foursquare** location data to solve. If you cannot think of an idea or a problem, here are some ideas to get you started:

1. In Module 3, we explored **New York City** and the city of **Toronto** and segmented and clustered their neighborhoods. Both cities are very diverse and are the financial capitals of their respective countries. One interesting idea would be to compare the neighborhoods of the two cities and determine how similar or dissimilar they are. Is New York City more like Toronto or Paris or some other multicultural city? I will leave it to you to refine this idea.

2. In a city of your choice, if someone is looking to open a **restaurant,** where would you recommend that they open it? Similarly, if a contractor is trying to start their own business, where would you recommend that they set up their office?

These are just a couple of many ideas and problems that can be solved using location data in addition to other datasets. No matter what you decide to do, make sure to provide sufficient justification of why you think what you want to do or solve is important and why would a client or a group of people be interested in your project.

2. INTRODUCCTION

I am currently living in the Long Island City neighborhood in New York City, which is in the Queens borough. I like my current neighborhood very much because of its multiculturalism, its venues and because of the density of venues. I have been living there for several years and really enjoy the surrounding.

Recently, I have been given a job offer in Milan, Italy. To work in this place, I must move to Brera Town in the Downtown of Milan. I have never been in Italy before. Therefore, it is a huge step for me to move there. I also have family. So, I want to make sure that there are enough high rated high schools in Brera Town. In addition to that, my family really like to eat Italian food.

I once lived in the Allapattah neighborhood in Miami, USA. I did not like it at all. So, before I move to Milan, Italy, I want to be sure, that the Brera Town neighborhood is more like the Long Island City neighborhood and not like Allapattah. If Brera Town is more like Long Island City and not like Allapattah, I will move to Milan, Italy. However, If Brera Town turns out to be more like Allapattah, which I did not like at all, I will decline the job offer and stay in Long Island City.

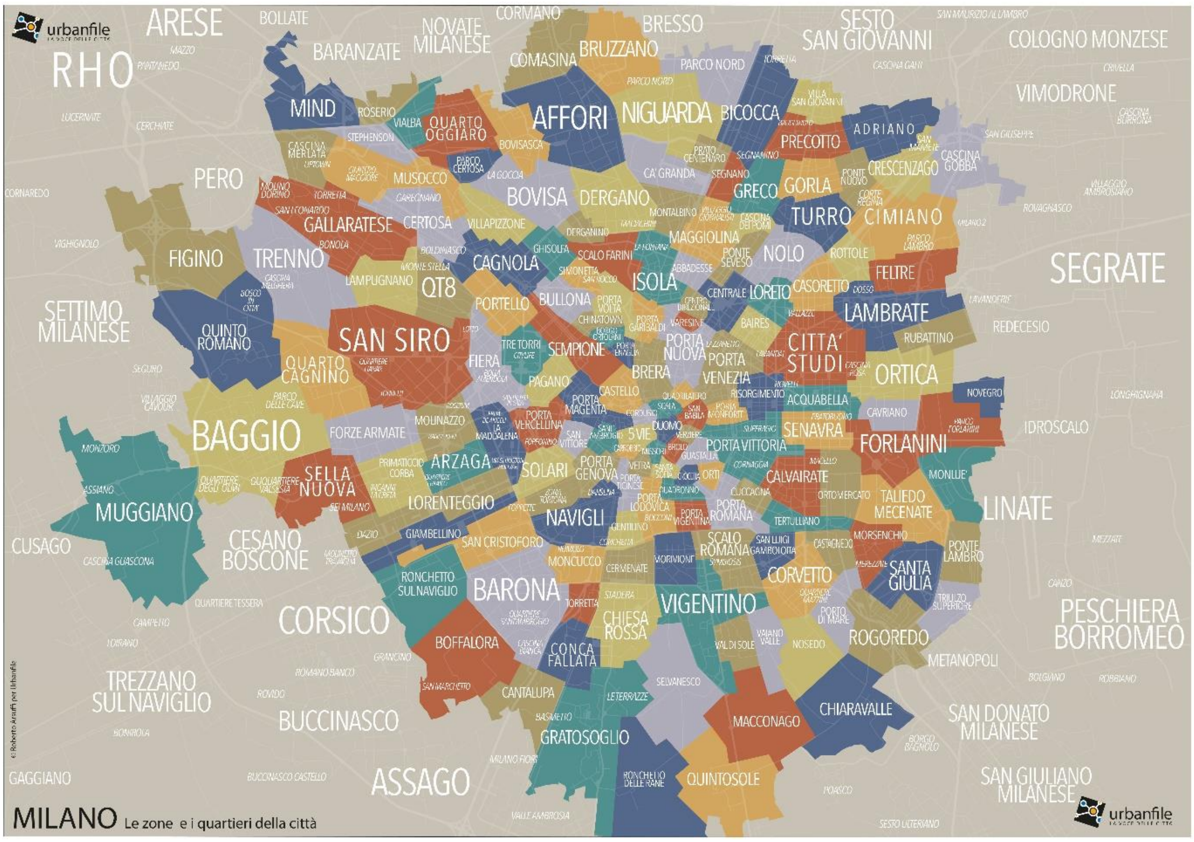
Going with the choice where I will be moving to Italy. I would like to start a business there and the question is: <u>Where would you recommend opening a new restaurant?</u>

3. BUSINESS PROBLEM

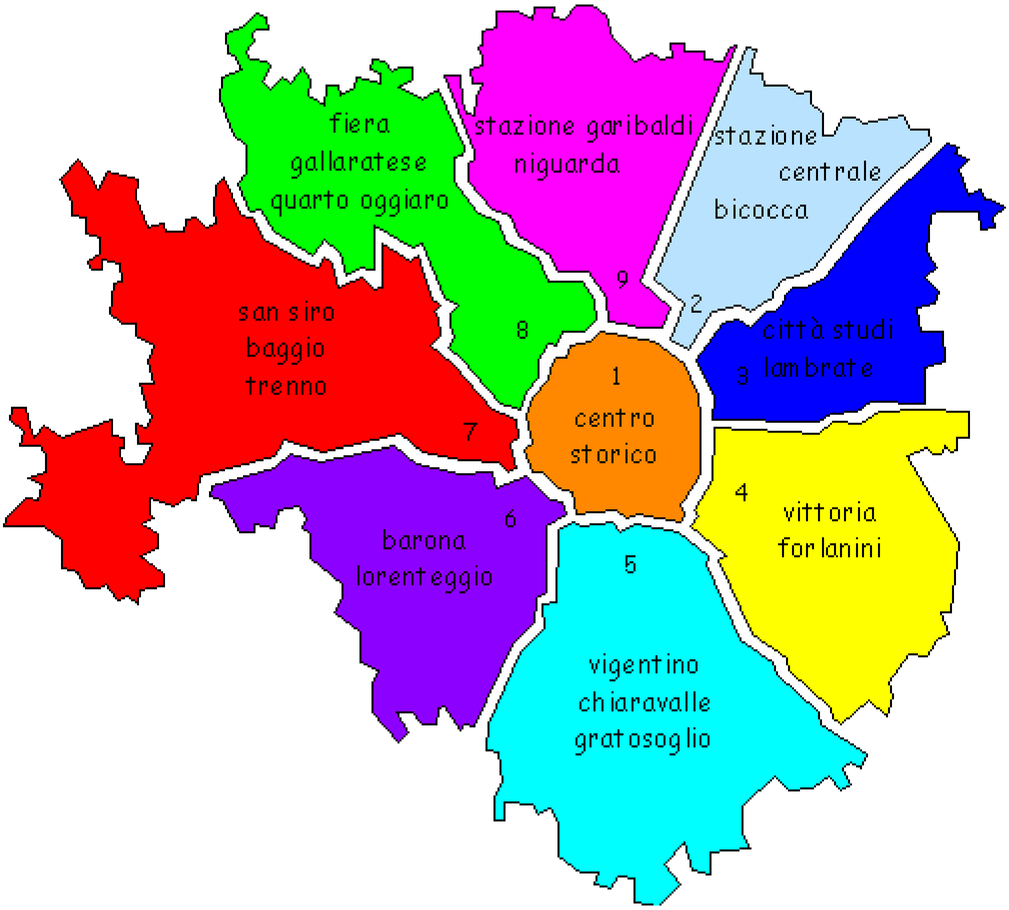
The city chosen to answer the initial question (Where would you recommend to opening a new restaurant?) is **Milan** a city in northern Italy, capital of Lombardy, and the second-most populous city in Italy after Rome. Its continuously built-up urban area, that stretches well beyond the boundaries of the administrative metropolitan city, is the fourth largest in the EU with 5.27 million inhabitants.

Milan is considered a leading alpha global city, with strengths in the field of the art, commerce, design, education, entertainment, fashion, finance, healthcare, media, services, research, and tourism. Its business district hosts Italy's stock exchange (*Italian: Borsa Italiana*), and the headquarters of national and international banks and companies. In terms of GDP, it has the second-largest economy among EU cities after Paris and is the wealthiest among EU non-capital cities. Milan is also considered part of the Blue Banana and one of the "Four Motors for Europe".

Let’s see how many neighbourhoods there are and how they are distributed:



As you can see there are many of them, so the town is also divided in districts (*municipi*):



After this short **presentation** , I suppose that the city of Milan is place with a great competition, especially, if you want to **open** a **restaurant** so I would like to help a possible stakeholder to understand better the town and the market with useful insights.

3.1. Target audience:

1. A business entrepreneur that wants open a new restaurant in Milan.
2. Business Analyst or Data Scientists, who wish to analyze the neighborhoods of Milan using python, Jupyter Notebook, and some machine learning techniques.
3. Someone curious about data that want to have an idea, how beneficial it is to open a restaurant and what are the pros and cons of this business.

3.2. Tools:

This project will make use of the programming language Python. Therefore, the following API's and libraries will be used:

1. **Foursquare API:**

   This API will be the primal source of data. The Foursquare compony gathers data of a huge variety of places and stores them in a database. The Foursquare API will be used to connect to the Foursquare database and get the needed information and data.

2. **Clustering techniques:**

   To compare different neighborhoods, this project will use different clustering approaches. By building models, we can therefore categorize the St. James Town neighborhood and compare it to other neighborhoods. K-Means will be one of the used algorithms.

3. **Libraries:**

   Pandas: For creating Data Frames to store / manipulate data and information.
   NumPy: As fundamental package for scientific computing.
   Folium: Python's visualization library to generate interactive leaflet maps.
   Scikit Learn: For developing different clustering / classification models.
   Matplotlib: For visualization of data.
   Geocoder: To retrieve locational Coordinates.
   JSON: To handle JSON files.

3.3. Data Section:

First, we need some information about the area of Milan such as borough\districts, population, latitude\longitude etc. Therefore, I think Wikipedia is the first place to look at:

https://it.wikipedia.org/wiki/Municipi_di_Milano

The boroughs are 9 with these coordinates:

	Borough	Name	Area(km2)	Population(31/12/2018)	Population_Density(km2)	Latitude	Longitude
0	1	Centro storico	967	98 531	10 189	45.471282	9.184999
1	2	Stazione Centrale, Gorla, Turro, Greco, Cresce...	1258	162 090	12 884	45.486117	9.203635
2	3	Città Studi, Lambrate, Venezia	1423	144 110	10 127	45.482506	9.241047
3	4	Vittoria, Forlanini	2095	161 551	7 711	45.431573	9.244738
4	5	Vigentino, Chiaravalle, Gratosoglio	2987	126 089	4 221	45.416987	9.238333
5	6	Barona, Lorenteggio	1828	151 291	8 276	45.440087	9.155924
6	7	Baggio, De Angeli, San Siro	3134	175 465	5 598	45.461244	9.089917
7	8	Fiera, Gallaratese, Quarto Oggiaro	2372	188 367	7 941	45.515925	9.140196
8	9	Stazione Garibaldi, Niguarda	2112	187 773	8 890	45.516888	9.191866

Now I need to find a list of all the **neighborhood** with the correspondent **borough.** Unfortunately, the Wikipedia tables are not up to date, so I found this paper from the official website of Milan:

https://www.pgt.comune.milano.it/sites/default/files/allegati/NIL_Intro.pdf

Municipio 1
1. Duomo
2. Brera
3. Giardini Porta Venezia
4. Guastalla
7. Magenta- San Vittore
8. Parco Sempione
(5. Vigentina)
(6. Ticinese)
(68. Pagano)
(69. Sarpi)

Municipio 2
10. Stazione Centrale - Ponte Seveso
16. Gorla - Precotto
17. Adriano
19. Padova - Turro - Crescenzago
(11. Isola)
(12. Maciachini-Maggiolina)
(13. Greco - Segnano)
(20. Loreto - Casoretto - NoLo)

Municipio 3
18. Cimiano - Rottole - Q.re Feltre
21. Buenos Aires - Porta Venezia - Porta Monforte
22. Città studi
23. Lambrate - Ortica
(20. Loreto)
(24. Parco Forlanini - Cavriano)

Municipio 4
25. Corsica
26. XXII Marzo
28. Umbria - Molise - Calvairate
29. Ortomercato
30. Taliedo - Morsenchio - Q.re Forlanini
31. Monluè - Ponte Lambro
32. Triulzo Superiore
33. Rogoredo - Santa Giulia
35. Lodi-Corvetto
(27. Porta Romana)

Municipio 5
5. Porta Vigentina - Porta Lodovica
6. Porta Ticinese - Conca del Naviglio
36. Scalo Romana
34. Chiaravalle
37. Morivione
38. Vigentino - Q.re Fatima
39. Quintosole
40. Ronchetto delle Rane
41. Gratosoglio - Q.re Missaglia - Q.re Terrazze
42. Stadera - Chiesa Rossa - Q.re Torretta - Conca Fallata
43. Tibaldi
85. Parco delle Abbazie
86. Parco dei Navigli
(47. Cantalupa)

Municipio 6
44. Porta Ticinese - Conchetta
45. Moncucco - San Cristoforo
46. Barona
47. Cantalupa
48. Ronchetto sul Naviglio - Q.re Lodovico il Moro
49. Giambellino
50. Porta Genova
52. Bande Nere
53. Lorenteggio
86. Parco dei Navigli
(51. Washington)

Municipio 7
51. Porta Magenta
54. Muggiano
55. Baggio - Q.re degli Olmi - Q.re Valsesia
56. Forze Armate
57. San Siro
58. De Angeli-Monte Rosa
60. Stadio - Ippodromi
61. Quarto Cagnino
62. Quinto Romano
63. Figino
87. Assiano
88. Parco Bosco in città
(68. Pagano)

Municipio 8
59. Tre Torri
64. Trenno
65. Q.re Gallaratese - Q.re San Leonardo - Lampugnano
66. QT8
67. Portello
68. Pagano
69. Sarpi
70. Ghisolfa
71. Villapizzone - Cagnola - Boldinasco
72. Maggiore - Musocco - Certosa
73. MIND - Cascina Triulza
74. Roserio
75. Stephenson
76. Quarto Oggiaro - Vialba - Musocco
(88. Parco Bosco in città)

Municipio 9
9. Porta Garibaldi - Porta Nuova
11. Isola
14. Niguarda - Ca' Granda - Prato Centenaro - Q.re Fulvio Testi
15. Bicocca
77. Bovisa
78. Farini
79. Dergano
80. Affori
81. Bovisasca
82. Comasina
83. Bruzzano
84. Parco Nord
(12. Maciachini-Maggiolina)
(13. Greco)

Scraping the pdf file was impossible, so I created and uploaded this dataset on GitHub: https://raw.githubusercontent.com/khalidalrifai/IBM-Data-Scientists-Professional/master/Milano_Municipi_NIL.csv

This is a sample:

	Num_Neighborhood	Neighborhood	Borough	Average_price_sm
0	17	Adriano	2	€ 2.800 /m²
1	80	Affori	9	€ 2.350 /m²
2	87	Assiano	7	€ 2.400 /m²
3	55	Baggio - Q.re degli Olmi - Q.re Valsesia	7	€ 2.400 /m²
4	52	Bande Nere	6	€ 3.857 /m²

Note: the information about average land price is taken from those two websites (national
reference points for the real estate market in Italy):
https://www.immobiliare.it/mercato-immobiliare/lombardia/milano/
https://www.mercato-immobiliare.info/lombardia/milano/milano.html

For the **final step**, I need to get the coordinates of every neighborhood. Fortunately, the statistics office of Milan created a very interesting portal about open data: https://dati.comune.milano.it/ and I found what I was looking for: a shape file (**geojson**).

https://dati.comune.milano.it/dataset/e8e765fc-d882-40b8-95d8-16ff3d39eb7c/resource/9c4e0776-56fc-4f3d-8a90-f4992a3be426/download/ds964_nil_wm.geojson

	ID_NIL	NIL	Valido_dal	Valido_al	Fonte	Shape_Length	Shape_Area	OBJECTID	geometry
0	48	RONCHETTO SUL NAVIGLIO - Q.RE LODOVICO IL MORO	05/02/2020	Vigente	Milano 2030 - PGT Approvato	8723.368714	2.406306e+06	89	POLYGON ((9.15422 45.43775, 9.15274 45.43887, ...
1	64	TRENNO	05/02/2020	Vigente	Milano 2030 - PGT Approvato	3309.998800	4.896921e+05	90	POLYGON ((9.10623 45.49016, 9.10591 45.49084, ...
2	67	PORTELLO	05/02/2020	Vigente	Milano 2030 - PGT Approvato	3800.750663	9.096022e+05	91	POLYGON ((9.15636 45.48785, 9.15495 45.48852, ...
3	81	BOVISASCA	05/02/2020	Vigente	Milano 2030 - PGT Approvato	7105.469715	1.578028e+06	92	POLYGON ((9.16803 45.52234, 9.16763 45.52272, ...
4	84	PARCO NORD	05/02/2020	Vigente	Milano 2030 - PGT Approvato	11741.717005	1.532331e+06	93	POLYGON ((9.20040 45.52848, 9.20028 45.52846, ...

After some steps of data cleaning and data preparation, the result is:

	Id	Neighborhood	Borough	Population(31/12/2018) Borough	Average Price(€/sm)	Latitude	Longitude	Geometry
0	48	Ronchetto Sul Naviglio - Q.Re Lodovico Il Moro	6	151 291	€ 2.563 /m²	45.438460	9.137260	POLYGON ((9.154221025150809 45.43775166985864,...
1	64	Trenno	8	188 367	€ 2.350 /m²	45.492822	9.101675	POLYGON ((9.10622707748691 45.4901620817868, 9...
2	67	Portello	8	188 367	€ 4.300 /m²	45.484490	9.153947	POLYGON ((9.156361803387014 45.48785426899354,...
3	81	Bovisasca	9	187 773	€ 2.000 /m²	45.517433	9.156731	POLYGON ((9.168034631377399 45.5223394320893, ...
4	84	Parco Nord	9	187 773	€ 6.800 /m²	45.523514	9.184235	POLYGON ((9.200403184652478 45.5284767228649, ...

Now I’m ready to use the **foursquare API**:
https://developer.foursquare.com/docs/places-api/

4. METHODOLOGY

4.1. Business Understanding

The aim of this project is to find the best neighborhood of Milan to open a new restaurant.

4.2. Analytical Approach

The total number of neighborhoods in Milan are 89 so we need to find a way to cluster them based on their similarities, that are the number and the kind of restaurant. Briefly, after some steps of Data Cleaning and Data Exploration, I will use a K-Means algorithm to extract the clusters, produce a map and make an argument on the result.

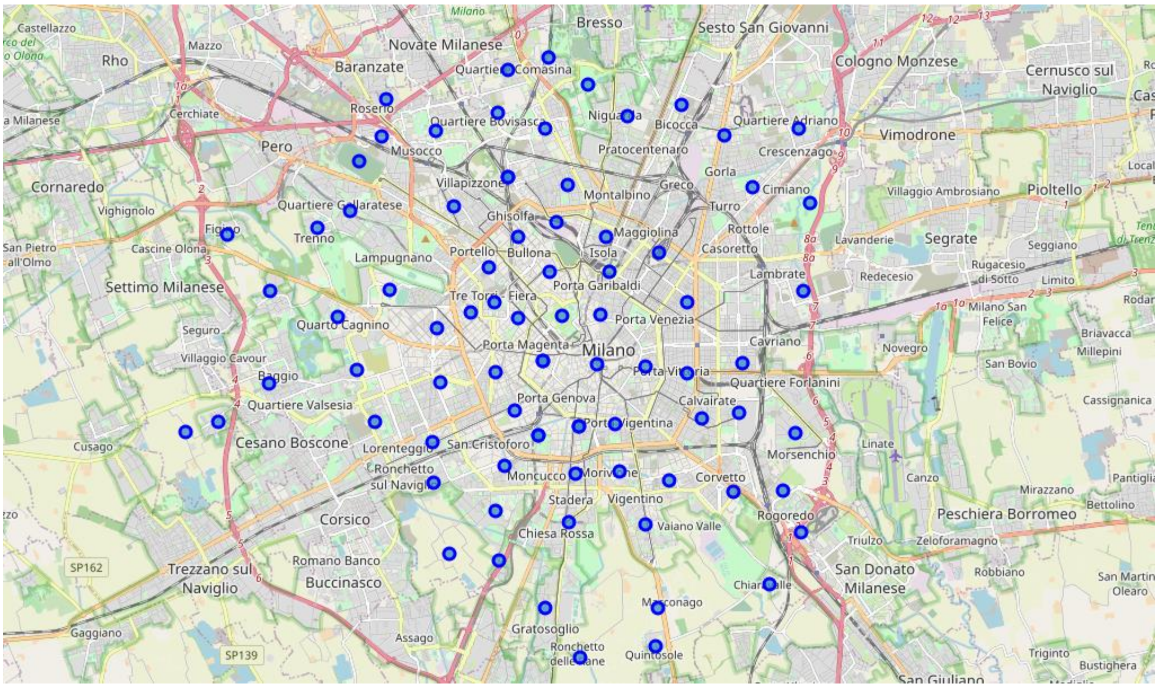
4.3. Data Exploration

To explore the data, I will use “Folium” a python library that can create interactive leaflet map using coordinate data. The code above is an example how to check the centroids of every neighborhood in Milan.

```
map_milan = folium.Map(location=[latitude, longitude], zoom_start=12)

# add markers to map
for lat, lng, borough, neighborhood in zip(df_milan_complete['Latitude'],
                                           df_milan_complete['Longitude'],
                                           df_milan_complete['Id'],
                                           df_milan_complete['Neighborhood']):
    label = '{}, {}'.format(neighborhood, borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_milan)

map_milan
```



Another interesting function is a GeoJSON map. Let's see:

```
m = folium.Map([latitude, longitude], zoom_start=12)

folium.GeoJson(milan_neighborhood_geodf,
    style_function=lambda x: {
        'color' : 'red',
        'opacity': 0.6,
        'fillColor' : 'green',
        }).add_to(m)

m
```

Now it's time to use the foursquare API (https://developer.foursquare.com/docs/api-reference/venues/explore/) to extract the venues of each neighborhood in Milan:

```
# create the API request URL
url = 'https://api.foursquare.com/v2/venues/explore?&section=food&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    lat,
    lng,
    radius,
    LIMIT)
```

```
rest_unique = milan_restaurants.groupby(['Venue',
                                         'Venue Latitude',
                                         'Venue Longitude',
                                         'Venue Category']).size().reset_index(name='Counts')
print(rest_unique.shape)
rest_unique.head(10)
```

(1994, 5)

]:

	Venue	Venue Latitude	Venue Longitude	Venue Category	Counts
0	"Carmen" (ristorante - pizzeria - grill)	45.440161	9.224682	Pizza Place	2
1	'A Tarantella	45.490889	9.233899	Pizza Place	1
2	100 Montaditos	45.446989	9.176994	Sandwich Place	2
3	100 Montaditos	45.453823	9.163722	Sandwich Place	2
4	100 Montaditos	45.522432	9.214766	Sandwich Place	2
5	13 Giugno	45.469227	9.215788	Restaurant	2
6	150up	45.490257	9.184895	Bistro	1
7	212 Hamburger & Delicious	45.454765	9.161157	Burger Joint	2
8	212 Rotisserie & Delicious	45.452686	9.201624	Fried Chicken Joint	5
9	22	45.474928	9.193852	Bistro	1

Unfortunately, if two centroids are too close together, I could extract duplicates venues (see the column "**Counts**"). To solve this problem, I will link a unique venue with the right neighborhood using his polygon ("**geometry**").

```
from shapely.geometry import shape, Point

rest_list = []

for ind1, rest in rest_unique.iterrows():
    point = Point(rest[["Venue Longitude"]].item(), rest[["Venue Latitude"]].item())
    # print(point)
    for ind2, neighborhood in df_milan_complete.iterrows():
        polygon = shape(neighborhood[["Geometry"]].item())
        if (polygon.contains(point)):
            # print("match with " + str(polygon))
            frame = {'Neighborhood': neighborhood[["Neighborhood"]].item(),
                     'Neighborhood Latitude': neighborhood[["Latitude"]].item(),
                     'Neighborhood Longitude': neighborhood[["Longitude"]].item(),
                     'Venue': rest[["Venue"]].item(),
                     'Venue Latitude': rest[["Venue Latitude"]].item(),
                     'Venue Longitude': rest[["Venue Longitude"]].item(),
                     'Venue Category': rest[["Venue Category"]].item()
                    }
            rest_list.append(frame)

cn = ['Neighborhood', 'Neighborhood Latitude', 'Neighborhood Longitude',
      'Venue', 'Venue Latitude', 'Venue Longitude', 'Venue Category']
milan_restaurants_unique = pd.DataFrame(rest_list, columns = cn)
milan_restaurants_unique.head()
```

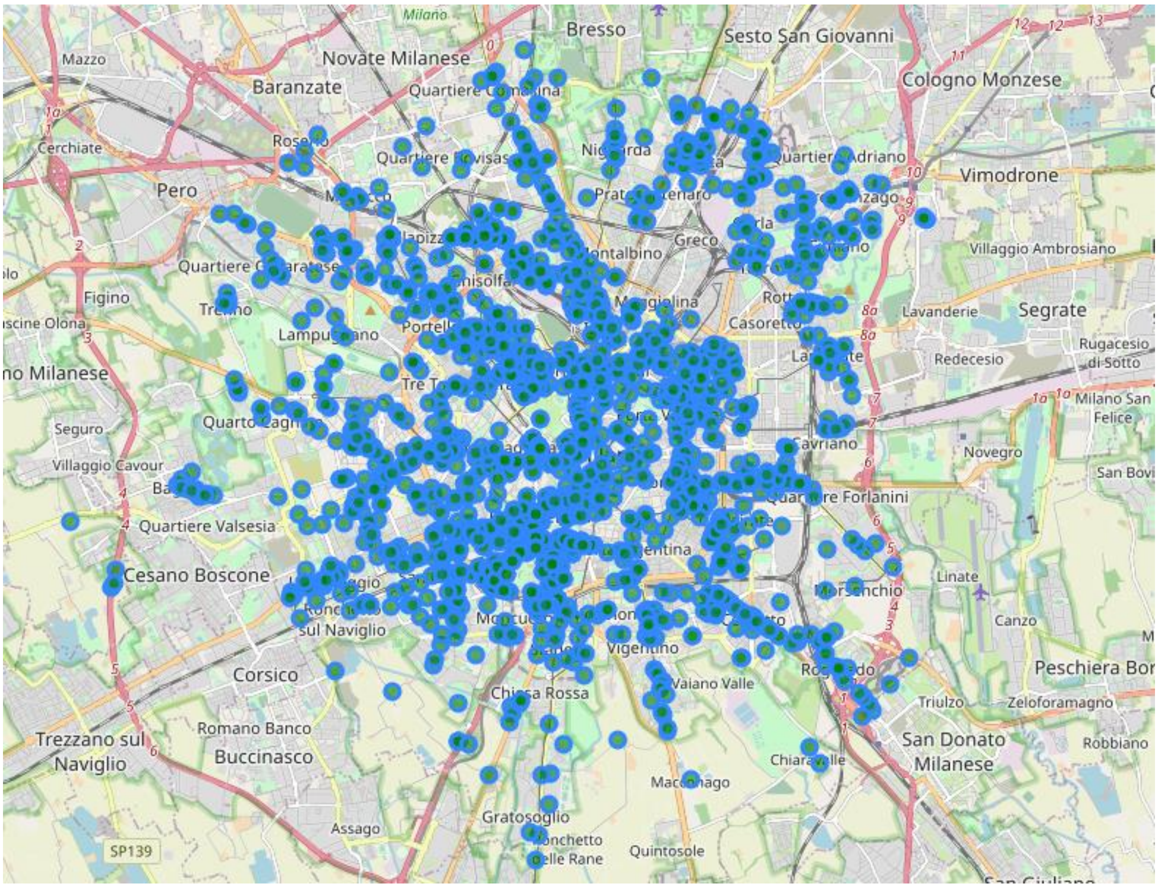
Let's compare the two datasets:

```
print(milan_restaurants.shape)
print(milan_restaurants_unique.shape)
```

```
(5139, 7)
(1852, 7)
```

As you can see, I removed a lot of duplicates.

Now, we can use "milan_restaurant_unique" dataset as input for a folium map:



Before proceeding, it could be a good idea to check what kind of venue are popular in Milan. Let's see:

```
venueDF = milan_restaurants_unique.groupby('Venue Category').size().reset_index(name='Counts')
venueDF.sort_values(by=['Counts'], ascending=False).head(10)
```

]:

	Venue Category	Counts
45	Italian Restaurant	396
17	Café	292
63	Pizza Place	264
66	Restaurant	92
46	Japanese Restaurant	80
8	Bakery	52
72	Seafood Restaurant	50
82	Sushi Restaurant	46
70	Sandwich Place	46
21	Chinese Restaurant	42

If we exclude Café and Bakery, Italian Restaurant and Pizza place are the most popular. Let's keep in mind and continue with our analysis.

4.4. Clustering:

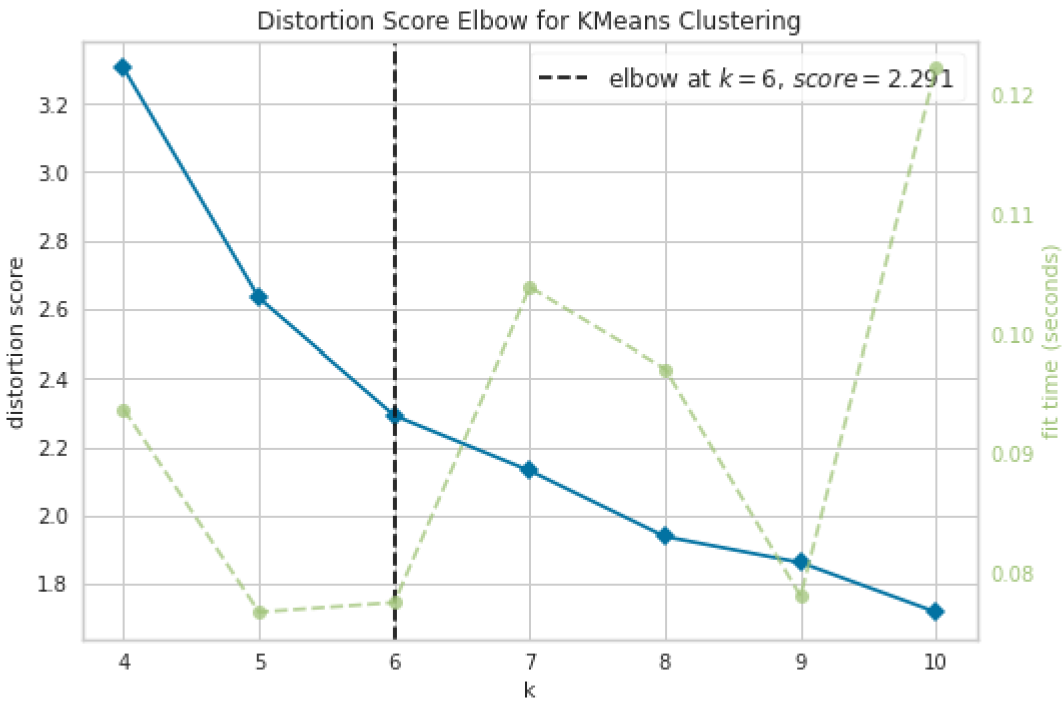
K-means clustering: a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity. Therefore, the first step is to identify the best "K" using a famous analytical approach: **the elbow method**.

```
!pip install yellowbrick
from sklearn.cluster import KMeans
from yellowbrick.cluster import KElbowVisualizer

milan_part_clustering = milan_grouped.drop('Neighborhood', 1)

# Instantiate the clustering model and visualizer
model = KMeans()
visualizer = KElbowVisualizer(model, k=(4,11))

visualizer.fit(milan_part_clustering)        # Fit the data to the visualizer
visualizer.poof()    # Draw/show/poof the data
```



From the plot up here, I can easily say that the best K is 6.

Finally, we can try to cluster the neighborhood based on the venue categories and use K-Means clustering. The 6 clusters are partitioned based on similar type of restaurants that belong to neighborhoods. To run the cluster, I have used the code snippet below.

```
# set number of clusters
kclusters = 6

milan_grouped_clustering = milan_grouped.drop('Neighborhood', 1)

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

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

And merge to obtain the final dataset:

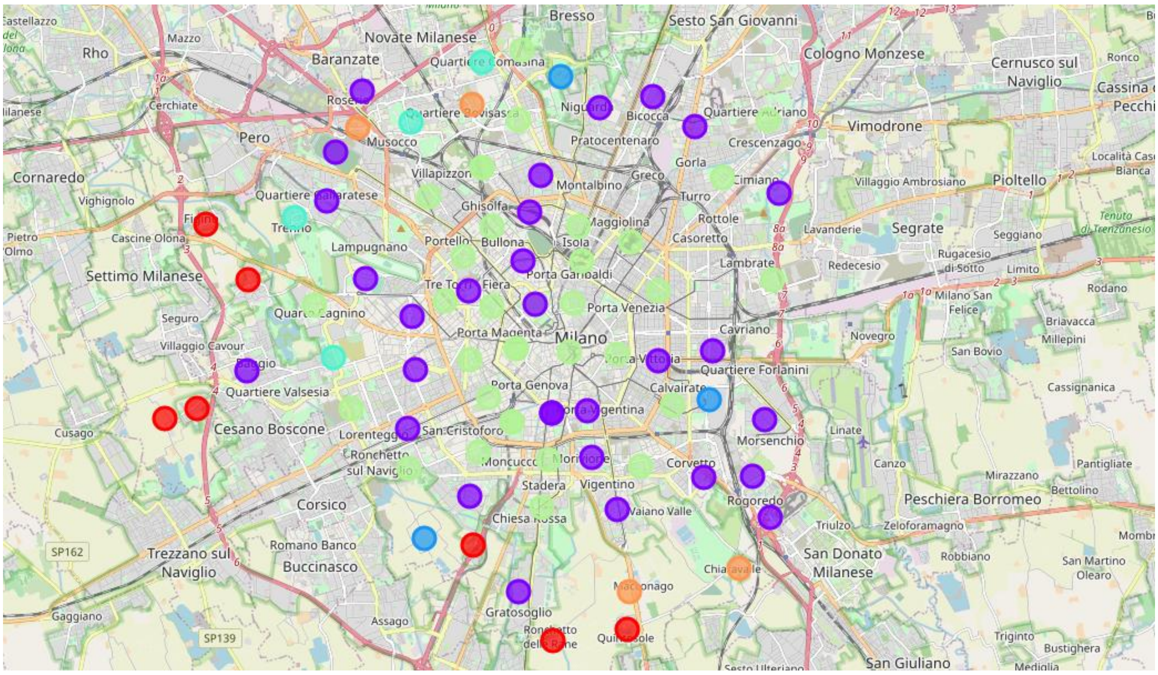
```
milan_merged = df_milan_complete.join(milan_venues_sorted.set_index('Neighborhood'), on='Neighborhood')
milan_merged['Cluster Labels'] = milan_merged['Cluster Labels'].fillna(0)
milan_merged['Cluster Labels'] = milan_merged['Cluster Labels'].astype(int)
milan_merged.drop(columns='Geometry', inplace=True)

milan_merged.head()
```

	Id	Neighborhood	Borough	Population(31/12/2018) Borough	Average Price(€/sm)	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	10th Most Common Venue
0	48	Ronchetto Sul Naviglio - Q.Re Lodovico Il Moro	6	151 291	€ 2.563 /m²	45.438460	9.137260	4	Italian Restaurant	Pizza Place	Food Court	Noodle House	Asian Restaurant	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria
1	64	Trenno	8	188 367	€ 2.350 /m²	45.492822	9.101675	3	Pizza Place	Sandwich Place	Spanish Restaurant	Asian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria
2	67	Portello	8	188 367	€ 4.300 /m²	45.484490	9.153947	4	Italian Restaurant	Japanese Restaurant	Seafood Restaurant	Mediterranean Restaurant	Restaurant	Pizza Place	Sandwich Place	Sushi Restaurant	Chinese Restaurant	Bistro
3	81	Bovisasca	9	187 773	€ 2.000 /m²	45.517433	9.156731	5	Italian Restaurant	Restaurant	Asian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria	Diner
4	84	Parco Nord	9	187 773	€ 6.800 /m²	45.523514	9.184235	2	Italian Restaurant	Sardinian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria	Diner	Asian Restaurant

5. RESULTS

Before to start to analyze all the clusters, let's look at the folium map:



As we can see, each cluster belong to a color with different characteristics. You can read the complete list above:

- Cluster 1 (Red):

```
milan_merged.loc[milan_merged['Cluster Labels'] == 0, milan_merged.columns[[1] + list(range(4, milan_merged.shape[1]))]]
```

	Neighborhood	Average Price(€/sm)	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	10th Most Common Venue
5	Figino	€ 2.000 /m²	45.491381	9.074376	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10	Quinto Romano	€ 2.250 /m²	45.479418	9.087541	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
20	Cantalupa	€ 3.717 /m²	45.421741	9.157204	0	Restaurant	Mediterranean Restaurant	Asian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria	Diner
21	Quintosole	€ 2.910 /m²	45.403412	9.204756	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
32	Assiano	€ 2.400 /m²	45.449368	9.061547	0	Restaurant	Mediterranean Restaurant	Asian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria	Diner
38	Muggiano	€ 2.200 /m²	45.451403	9.071630	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
43	Ronchetto Delle Rane	€ 4.350 /m²	45.401107	9.181961	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

- Cluster 2 (Purple):

```
milan_merged.loc[milan_merged['Cluster Labels'] == 1, milan_merged.columns[[1] + list(range(4, milan_merged.shape[1]))]]
```

	Neighborhood	Average Price(€/sm)	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	10th Most Common Venue
9	Stadio - Ippodromi	€ 3.265 /m²	45.479641	9.123833	1	Seafood Restaurant	Pizza Place	Italian Restaurant	Mediterranean Restaurant	Food Truck	Diner	Burger Joint	Japanese Restaurant	Restaurant	Sushi Restaurant
13	San Siro	€ 3.150 /m²	45.471382	9.138358	1	Chinese Restaurant	Pizza Place	Italian Restaurant	Sushi Restaurant	Trattoria/Osteria	Food Truck	Sardinian Restaurant	Sandwich Place	Seafood Restaurant	Bistro
17	Farini	€ 5.652 /m²	45.493963	9.174605	1	Italian Restaurant	Pizza Place	Restaurant	Breakfast Spot	Seafood Restaurant	Indian Restaurant	Chinese Restaurant	Noodle House	Diner	Piadineria
22	Parco Sempione	€ 5.800 /m²	45.474131	9.176251	1	Spanish Restaurant	Sardinian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria	Diner	Asian Restaurant
23	Barona	€ 3.250 /m²	45.432353	9.156192	1	Food Court	Italian Restaurant	Pizza Place	Japanese Restaurant	Trattoria/Osteria	Asian Restaurant	Breakfast Spot	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria
25	Gorla - Precotto	€ 2.800 /m²	45.512660	9.225630	1	Pizza Place	Italian Restaurant	Breakfast Spot	Restaurant	Japanese Restaurant	Seafood Restaurant	Puglia Restaurant	Trattoria/Osteria	Chinese Restaurant	Food Truck
26	Niguarda - Ca' Granda - Prato Centenaro - Q.Re...	€ 2.550 /m²	45.516696	9.196117	1	Pizza Place	Restaurant	Sushi Restaurant	Italian Restaurant	Korean Restaurant	Asian Restaurant	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant
27	Triulzo Superiore	€ 2.626 /m²	45.427941	9.249243	1	Fast Food Restaurant	Pizza Place	Restaurant	Japanese Restaurant	Trattoria/Osteria	Diner	Steakhouse	Breakfast Spot	Kebab Restaurant	Vegetarian / Vegan Restaurant
28	Taliedo - Morsenchio - Q.Re Forlanini	€ 2.950 /m²	45.449146	9.247377	1	Italian Restaurant	Pizza Place	Fast Food Restaurant	Breakfast Spot	Greek Restaurant	Asian Restaurant	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria
29	Porta Ticinese - Conca Del Naviglio	€ 6.150 /m²	45.450475	9.181311	1	Italian Restaurant	Pizza Place	Japanese Restaurant	Vegetarian / Vegan Restaurant	Restaurant	Bistro	Sushi Restaurant	Diner	Indian Restaurant	Burger Joint

- Cluster 3 (Light Blue):

```
milan_merged.loc[milan_merged['Cluster Labels'] == 2, milan_merged.columns[[1] + list(range(4, milan_merged.shape[1]))]]
```

	Neighborhood	Average Price(€/sm)	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	10th Most Common Venue
4	Parco Nord	€ 6.800 /m²	45.523514	9.184235	2	Italian Restaurant	Sardinian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria	Diner	Asian Restaurant
36	Ortomercato	€ 4.005 /m²	45.453417	9.230270	2	Italian Restaurant	Pizza Place	Asian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria	Diner
75	Parco Dei Navigli	€ 1.800 /m²	45.423321	9.141989	2	Italian Restaurant	Sardinian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria	Diner	Asian Restaurant

- Cluster 4 (Cyan):

```
milan_merged.loc[milan_merged['Cluster Labels'] == 3, milan_merged.columns[[1] + list(range(4, milan_merged.shape[1]))]]
```

	Neighborhood	Average Price(€/sm)	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	10th Most Common Venue
1	Trenno	€ 2.350 /m²	45.492822	9.101675	3	Pizza Place	Sandwich Place	Spanish Restaurant	Asian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria
6	Quarto Oggiaro - Vialba - Musocco	€ 1.700 /m²	45.513636	9.137731	3	Pizza Place	Spanish Restaurant	Asian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria	Diner
14	Comasina	€ 1.750 /m²	45.526441	9.159969	3	Pizza Place	Spanish Restaurant	Asian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria	Diner
48	Forze Armate	€ 2.700 /m²	45.462489	9.113830	3	Pizza Place	Fast Food Restaurant	Asian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria	Diner

- Cluster 5 (Green):

```
milan_merged.loc[milan_merged['Cluster Labels'] == 4, milan_merged.columns[[1] + list(range(4, milan_merged.shape[1]))]]
```

	Neighborhood	Average Price(€/sm)	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	10th Most Common Venue
0	Ronchetto Sul Naviglio - Q.Re Lodovico Il Moro	€ 2.563 /m²	45.438460	9.137260	4	Italian Restaurant	Pizza Place	Food Court	Noodle House	Asian Restaurant	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria
2	Portello	€ 4.300 /m²	45.484490	9.153947	4	Italian Restaurant	Japanese Restaurant	Seafood Restaurant	Mediterranean Restaurant	Restaurant	Pizza Place	Sandwich Place	Sushi Restaurant	Chinese Restaurant	Bistro
7	Isola	€ 5.550 /m²	45.490894	9.189617	4	Italian Restaurant	Pizza Place	Bistro	Ramen Restaurant	Restaurant	Seafood Restaurant	Chinese Restaurant	Burger Joint	Japanese Restaurant	Vegetarian / Vegan Restaurant
8	Quarto Cagnino	€ 2.241 /m²	45.473740	9.108096	4	Italian Restaurant	Pizza Place	Sushi Restaurant	Asian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria
11	Duomo	€ 7.100 /m²	45.463707	9.186948	4	Italian Restaurant	Pizza Place	Sandwich Place	Restaurant	Bistro	Sushi Restaurant	Burger Joint	Japanese Restaurant	American Restaurant	Sardinian Restaurant
12	Guastalla	€ 7.300 /m²	45.463219	9.201891	4	Italian Restaurant	Pizza Place	Seafood Restaurant	Indian Restaurant	Bistro	Sandwich Place	Fried Chicken Joint	Japanese Restaurant	Restaurant	Falafel Restaurant
15	Tibaldi	€ 2.750 /m	45.440348	9.180459	4	Italian Restaurant	Pizza Place	Japanese Restaurant	Sushi Restaurant	Food Court	Breakfast Spot	Diner	Steakhouse	Chinese Restaurant	Bistro

- Cluster 6 (Orange):

```
milan_merged.loc[milan_merged['Cluster Labels'] == 5, milan_merged.columns[[1] + list(range(4, milan_merged.shape[1]))]]
```

]:

	Neighborhood	Average Price(€/sm)	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	10th Most Common Venue
3	Bovisasca	€ 2.000 /m²	45.517433	9.156731	5	Italian Restaurant	Restaurant	Asian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria	Diner
19	Stephenson	€ 3.079 /m²	45.512246	9.121394	5	Italian Restaurant	Steakhouse	Restaurant	Asian Restaurant	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria	Diner
50	Chiaravalle	€ 2.700 /m²	45.416749	9.239611	5	Italian Restaurant	Restaurant	Asian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria	Diner
51	Parco Delle Abbazie	€ 4.300 /m²	45.411618	9.205639	5	Italian Restaurant	Restaurant	Japanese Restaurant	Asian Restaurant	Steakhouse	Breakfast Spot	Trattoria/Osteria	Kebab Restaurant	Vegetarian / Vegan Restaurant	Piadineria

Here we are at the end of the analysis, I tried to set up a realistic data-analysis scenario using several different ways such as: web scraping on Wikipedia, open data from public administration (Mayor of Milan), some powerful python libraries e.g. Folium and GeoPandas, Foursquare API, etc.

Now we can make some argument about the clusters. Let's see what we have found:

1. The most common venues in Milan are Italian Restaurant and Pizza Place.
2. Cluster 3 and 5 don't have an Italian Restaurant.
3. From the geographical representation of the clusters, Comanasina and Quarto Oggiaro, seems a good place open an Italian Restaurant. Also, the land price isn't so high.
4. If our stakeholder thinks that there are too much Italian Restaurant, it can also be suggested that Assiano (cluster 5) could be a great area to open a Vegan\Vegetarian restaurant because of low profile and land price.

6. CONCLUSION

As the analysis is performed on small set of data, we can achieve better results by increasing the neighborhood information (see the next chapter). Anyway Milan is an international city with many different types of new restaurant business to offer and I think we have gone through the process of identifying the business problem, specifying the data required, clean the datasets, performing a machine learning algorithm using k-means clustering and providing some useful tips to our stakeholder.

7. NEXT DEVELOPMENTS

Next steps I recommend would be:

- Use a different Venue API with more data. Unfortunately, foursquare isn't famous in Italy. Mostly users prefer Google Maps or Facebook.
- Find and use updated demographics data about Milan's Neighborhood.
- Try a Neighborhood-Based Clustering.