Exploring Milan districts: a simple project of data science.

Fabio Marco Monetti

April 28, 2020

Chapter 1

Introduction

Taking inspiration from the un-graded laboratories that were offered by the course and from one of the ideas suggested in the last week of this course, I decided to explore and inspect the boroughs of my own city of birth: Milan, in Italy. In this project I will explore the main boroughs of the city and see what kind of venues are the most popular. This idea occurred because I want to remember my beautiful city as it was before the spreading of this awful virus called Covid-19 that is affecting our lives and will, for the foreseeable future, change how we conduct daily life and the relationship that everyone has with the city in which they live.

In recent years, after the Global Exposition of 2015, Milan has been one of the most visited cities in Europe; therefore, plenty of data is available on Foresquare about the main venues that one could visit.

Moreover, since the numbers of tourists coming in the city have been steadily growing up and it seemed that they would never going to drop, opening a new restaurant in Milan, or, for what it matters, any kind of venues that one could think of, might seem a bright idea to exploit this growth. It is true that competition is high in Milan, therefore it is advisable to run some data analysis in order to find out which parts of the city are developing and where is the best place to open a new restaurant.

In this project I will go through the process of making such a decision.

Chapter 2

Data

2.1 Data acquisition

I will use “Municipalities of Milan” () wiki page to get information about the boroughs in Milan. This page has a table that displays plenty of data about the boroughs; it also includes:

- boroughs;
- name;
- area;
- population;
- population density;
- quartieri (we could translate as "districts").

I will use a query in pgeocode and the geopy package to retrieve all the informations about the coordinates of those boroughs.

To get location and information about various venues in Milan I will use the Foursquare explore API. Using the borough’s postal codes, from Foursquare API (), I retrieved the following informations about the venues:

- name;
- category;
- latitude;
- longitude.

2.2 Data cleaning

Firstly, I scraped the Wikipedia page in order to obtain the data about the municipalities (or boroughs) of Milan. I used the BeautifulSoup package to scrape the data.

In extracting data from the table I retrieved also data that were not considered useful for the purpose of this project (though they can turn out to be of the most importance if this study is to be followed and implemented with further studies - so of course nothing has been permanently deleted). Therefore, at the end of the scraping project, the extracted dataframe presented nine rows (one for each borough) and five columns, as you can see in the 2.1. These data needed to be cleansed, too, because there were inconsistencies about the separator for decimals and thousands (sometimes a comma, sometimes a point, for both). The table you see is the rightly formatted one.

Secondly, I imported pgeocode to retrieve data about the latitude and longitude of the boroughs. In order to do so, I needed the postal code of those boroughs, which unfortunately was not part of the Wikipedia table; so, I went on the webpage of "Comune di Milano" (), and I was able to use those data to run a query on pgeocode with the postal codes and retrieve the coordinates of the boroughs.

I then added those data to the previous dataframe, see 2.2. Lastly, thanks to the Foursquare API, I retrieved information about the venues present in the boroughs of Milan. I looked for at most a hundred places in a

Figure 2.1

	Borough	Name	Area	Population	Density	District
0	1	Centro storico	9.67	96315	11074	Brera
1	2	Stazione Centrale	12.58	153109	13031	Adriano
2	3	Città Studi	14.23	141229	10785	Casoretto
3	4	Porta Vittoria	20.95	156369	8069	Acquabella
4	5	Vigentino	29.87	123779	4487	Basmetto
5	6	Barona	18.28	149000	8998	Arzaga
6	7	Baggio	31.34	170814	6093	Assiano
7	8	Fiera	23.72	181669	8326	Boldinasco
8	9	Porta Garibaldi	21.12	181598	9204	Affori

Figure 2.2

	Borough	Name	Area	Population	Density	District	Latitude	Longitude
0	1	Centro storico	9.67	96315	11074	Brera	45.5024	9.200175
1	2	Stazione Centrale	12.58	153109	13031	Adriano	45.5077	9.218700
2	3	Città Studi	14.23	141229	10785	Casoretto	45.4823	9.215250
3	4	Porta Vittoria	20.95	156369	8069	Acquabella	45.4313	9.217200
4	5	Vigentino	29.87	123779	4487	Basmetto	45.4122	9.180500
5	6	Barona	18.28	149000	8998	Arzaga	45.4371	9.171950
6	7	Baggio	31.34	170814	6093	Assiano	45.5049	9.158000
7	8	Fiera	23.72	181669	8326	Boldinasco	45.5168	9.163000
8	9	Porta Garibaldi	21.12	181598	9204	Affori	45.4643	9.189500

Figure 2.3

	name	categories	lat	lng
0	Palestra McFIT	Gym	45.504724	9.199265
1	Pasticceria Martesana	Dessert Shop	45.495824	9.203095
2	Nisida	Pizza Place	45.507636	9.203686
3	BB Hotels Residenza Bicocca	Bed & Breakfast	45.499852	9.198630
4	Il Borghetto	Steakhouse	45.501415	9.209785

```
print('{} venues were returned by Foursquare.'.format(nearby_venues.shape[0]))
```

```
100 venues were returned by Foursquare.
```

Figure 2.4

Shape of the Venues Dataframe: (535, 7)

Out[64]:

	Name	Latitude	Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Centro storico	45.5024	9.200175	Palestra McFIT	45.504724	9.199265	Gym
1	Centro storico	45.5024	9.200175	BB Hotels Residenza Bicocca	45.499852	9.198630	Bed & Breakfast
2	Centro storico	45.5024	9.200175	Ristorante Sirenella	45.500322	9.198919	Seafood Restaurant
3	Centro storico	45.5024	9.200175	Nisida	45.507636	9.203686	Pizza Place

radius of 1 km around every center of borough. The API returned a JSON file that was then formatted to become a data frame. For example, for the first borough, you can see what I did in 2.3. And the complete data frame looked like you can see in 2.4.