
Milan Sweet
Coffee | Ice Cream | Chocolate | Cupcake | Pastry | Dessert

Capstone Project - The Battle of Neighborhoods

- It is the final excercise of IBM Data Science Professional Certificate training on Coursera in order to obtain the professional certification «IBM Data Science»
- Foursquare integration with data limitations - Free developer account

City Background

Milan is a city in northern Italy, capital of Lombardy, and the second-most populous city in Italy after Rome. Milan served as the capital of the Western Roman Empire, the Duchy of Milan and the Kingdom of Lombardy-Venetia. The city proper has a population of about 1.4 million while its metropolitan city has 3.26 million inhabitants. 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. The population within the wider Milan metropolitan area, also known as Greater Milan, is estimated at 8.2 million, making it by far the largest metropolitan area in Italy and the 4th largest in the EU. 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.

Business Opportunity

- Milan is one of the best world cities to invest money, it has been recognized as one of the world's four fashion capitals thanks to several international events and fairs, including Milan Fashion Week and the Milan Furniture Fair, which are currently among the world's biggest in terms of revenue, visitors and growth. It hosted the Universal Exposition in 1906 and 2015. The city hosts numerous cultural institutions, academies and universities, with 11% of the national total enrolled students. Milan is the destination of 8 million overseas visitors every year, attracted by its museums and art galleries that include some of the most important collections in the world, including major works by Leonardo da Vinci. The city is served by many luxury hotels and is the fifth-most starred in the world by Michelin Guide. The city is home to two of Europe's most successful football teams, A.C. Milan and F.C. Internazionale, and one of Europe's main basketball teams, Olimpia Milano. Milan will host the 2026 Winter Olympics together with Cortina d'Ampezzo.
- During the day and specifically during breakfast time (07:30 – 10:00) and lunch time (12:00 – 14:30) the areas with higher population density are struggling to manage the flow of people who want to treat themselves to a dessert.Within this scenario where the city continues to face a continuous population increase, we will go to analyze the possibility of opening a handicraft shop of sweets and coffee.
- Milan has 9 town halls and our focus will be on Centro storico, Stazione centrale, Città studi, Porta vittoria, Vigentino, Barona, Baggio and Porta garibaldi

Data Collection sources

- Wikipedia: list of all townhalls in Milan
- Geopy: locating the coordinates of addresses, cities, countries and landmarks
- Mercato-immobiliare.info: Average price (RealEstate)
- Foursquare: Data of existing points of interests

Townhall list

	Townhall	Description	Area	Population	Density	Name
0	1	Centro storico	9.67	96,315	11,074	Centro storico
1	2	Stazione Centrale, Gorla, Turro, Greco, Cresce...	12.58	153.109	13,031	Stazione centrale
2	3	Città Studi, Lambrate, Porta Venezia	14.23	141,229	10,785	Città studi
3	4	Porta Vittoria, Forlanini	20.95	156.369	8,069	Porta vittoria
4	5	Vigentino, Chiaravalle, Gratosoglio	29.87	123,779	4,487	Vigentino
5	6	Barona, Lorenteggio	18.28	149,000	8,998	Barona
6	7	Baggio, De Angeli, San Siro	31.34	170,814	6,093	Baggio
7	8	Fiera, Gallaratese, Quarto Oggiaro	23.72	181,669	8,326	Fiera
8	9	Porta Garibaldi, Niguarda	21.12	181,598	9,204	Porta garibaldi

Webscraping from wikipedia to collect information into dataframe

Data Preparation - Wikipedia

In [3]:
```
response = requests.get('https://en.wikipedia.org/wiki/Zones_of_Milan').text
soup = BeautifulSoup(response,'lxml')
table = soup.find('table', {'class':'wikitable sortable'})
```

1

In [4]:
```
table_rows = table.find_all('tr')

res = []
for tr in table_rows:
    td = tr.find_all('td')
    row = [tr.text.strip() for tr in td if tr.text.strip()]
    if row:
        res.append(row)


df = pd.DataFrame(res, columns=["Townhall", "Description", "Area", "Population","Density", "Districts"])
df
```

Out[4]:

	Townhall	Description	Area	Population	Density	Districts
0	1	Centro storico	9.67	96,315	11,074	Brera, Centro Storico, Conca del Naviglio, Gua...
1	2	Stazione Centrale, Gorla, Turro, Greco, Cresce...	12.58	153.109	13,031	Adriano, Crescenzago, Gorla, Greco, Loreto, Ma...
2	3	Città Studi, Lambrate, Porta Venezia	14.23	141,229	10,785	Casoretto, Cimiano, Città Studi, Dosso, Lambra...
3	4	Porta Vittoria, Forlanini	20.95	156.369	8,069	Acquabella, Calvairate, Castagnedo, Cavriano, ...
4	5	Vigentino, Chiaravalle, Gratosoglio	29.87	123,779	4,487	Basmetto, Cantalupa, Case Nuove, Chiaravalle, ...
5	6	Barona, Lorenteggio	18.28	149,000	8,998	Arzaga, Barona, Boffalora, Cascina Bianca, Con...
6	7	Baggio, De Angeli, San Siro	31.34	170,814	6,093	Assiano, Baggio, Figino, Fopponino, Forze Arma...
7	8	Fiera, Gallaratese, Quarto Oggiaro	23.72	181,669	8,326	Boldinasco, Bullona, Cagnola, Campo dei Fiori,...
8	9	Porta Garibaldi, Niguarda	21.12	181,598	9,204	Affori, Bicocca, Bovisa, Bovisasca, Bruzzano, ...

In [6]:
```
df_Townhall['Name'] = df_Townhall.Description.str.split(',').str[0]
df_Townhall['Name'] = df_Townhall.Name.str.capitalize()
df_Townhall
```

2

Out[6]:

	Townhall	Description	Area	Population	Density	Name
0	1	Centro storico	9.67	96,315	11,074	Centro storico
1	2	Stazione Centrale, Gorla, Turro, Greco, Cresce...	12.58	153.109	13,031	Stazione centrale
2	3	Città Studi, Lambrate, Porta Venezia	14.23	141,229	10,785	Città studi
3	4	Porta Vittoria, Forlanini	20.95	156.369	8,069	Porta vittoria
4	5	Vigentino, Chiaravalle, Gratosoglio	29.87	123,779	4,487	Vigentino
5	6	Barona, Lorenteggio	18.28	149,000	8,998	Barona
6	7	Baggio, De Angeli, San Siro	31.34	170,814	6,093	Baggio
7	8	Fiera, Gallaratese, Quarto Oggiaro	23.72	181,669	8,326	Fiera
8	9	Porta Garibaldi, Niguarda	21.12	181,598	9,204	Porta garibaldi

Townhall Coordinates

	Townhall	Description	Area	Population	Density	Name	Latitude	Longitude
0	1	Centro storico	9.67	96,315	11,074	Centro storico	45.467281	9.185962
1	2	Stazione Centrale, Gorla, Turro, Greco, Cresce...	12.58	153.109	13,031	Stazione centrale	45.499990	9.218999
2	3	Città Studi, Lambrate, Porta Venezia	14.23	141,229	10,785	Città studi	45.489800	9.241031
3	4	Porta Vittoria, Forlanini	20.95	156.369	8,069	Porta vittoria	45.446205	9.239343
4	5	Vigentino, Chiaravalle, Gratosoglio	29.87	123,779	4,487	Vigentino	45.419846	9.198153
5	6	Barona, Lorenteggio	18.28	149,000	8,998	Barona	45.437929	9.145648
6	7	Baggio, De Angeli, San Siro	31.34	170,814	6,093	Baggio	45.469598	9.114757
7	8	Fiera, Gallaratese, Quarto Oggiaro	23.72	181,669	8,326	Fiera	45.500015	9.122774
8	9	Porta Garibaldi, Niguarda	21.12	181,598	9,204	Porta garibaldi	45.507704	9.179410

Geopy makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources. Thanks to Geopy we collect Latitude and Longitude for each Townhall.

Data Preparation - Geopy

```
In [7]: from geopy.geocoders import Nominatim
        geolocator = Nominatim()
        location = geolocator.geocode("Milano, MI, Lom, Italia")
        address = []
        coord = []
        address = "Municipio "+ df['Townhall']+", Milano, MI, Lom, Italy"
        coord = address.apply(geolocator.geocode).apply(lambda x: (x.latitude, x.longitude))
        df_Townhall['Coordinates'] = coord
        df_Townhall
```

1

Out[7]:

	Townhall	Description	Area	Population	Density	Name	Coordinates
0	1	Centro storico	9.67	96,315	11,074	Centro storico	(45.467280599999995, 9.185962010120925)
1	2	Stazione Centrale, Gorla, Turro, Greco, Cresce...	12.58	153.109	13,031	Stazione centrale	(45.4999899, 9.218998626052084)
2	3	Città Studi, Lambrate, Porta Venezia	14.23	141,229	10,785	Città studi	(45.489799500000004, 9.241030886135526)
3	4	Porta Vittoria, Forlanini	20.95	156.369	8,069	Porta vittoria	(45.446205, 9.239343264566283)
4	5	Vigentino, Chiaravalle, Gratosoglio	29.87	123,779	4,487	Vigentino	(45.4198458, 9.198152549352766)
5	6	Barona, Lorenteggio	18.28	149,000	8,998	Barona	(45.43792895, 9.145648151241897)
6	7	Baggio, De Angeli, San Siro	31.34	170,814	6,093	Baggio	(45.469598000000005, 9.114757320322848)
7	8	Fiera, Gallaratese, Quarto Oggiaro	23.72	181,669	8,326	Fiera	(45.50001535, 9.122774225147062)
8	9	Porta Garibaldi, Niguarda	21.12	181,598	9,204	Porta garibaldi	(45.5077041, 9.17941004494827)

```
In [8]: df_Townhall[['Latitude', 'Longitude']] = df_Townhall['Coordinates'].apply(pd.Series)
```

```
In [9]: df_Townhall = df_Townhall.drop(columns=['Coordinates'])
```

2

```
In [10]: df_Townhall
```

Out[10]:

	Townhall	Description	Area	Population	Density	Name	Latitude	Longitude
0	1	Centro storico	9.67	96,315	11,074	Centro storico	45.467281	9.185962
1	2	Stazione Centrale, Gorla, Turro, Greco, Cresce...	12.58	153.109	13,031	Stazione centrale	45.499990	9.218999
2	3	Città Studi, Lambrate, Porta Venezia	14.23	141,229	10,785	Città studi	45.489800	9.241031
3	4	Porta Vittoria, Forlanini	20.95	156.369	8,069	Porta vittoria	45.446205	9.239343
4	5	Vigentino, Chiaravalle, Gratosoglio	29.87	123,779	4,487	Vigentino	45.419846	9.198153
5	6	Barona, Lorenteggio	18.28	149,000	8,998	Barona	45.437929	9.145648
6	7	Baggio, De Angeli, San Siro	31.34	170,814	6,093	Baggio	45.469598	9.114757
7	8	Fiera, Gallaratese, Quarto Oggiaro	23.72	181,669	8,326	Fiera	45.500015	9.122774
8	9	Porta Garibaldi, Niguarda	21.12	181,598	9,204	Porta garibaldi	45.507704	9.179410

Real Estate average price

	Townhall	Description	Area	Population	Density	Name	Latitude	Longitude	Price
0	1	Centro storico	9.67	96,315	11,074	Centro storico	45.467281	9.185962	€ 7.100 /m²
1	2	Stazione Centrale, Gorla, Turro, Greco, Cresce...	12.58	153.109	13,031	Stazione centrale	45.499990	9.218999	€ 5.000 /m²
2	3	Città Studi, Lambrate, Porta Venezia	14.23	141,229	10,785	Città studi	45.489800	9.241031	€ 4.500 /m²
3	4	Porta Vittoria, Forlanini	20.95	156.369	8,069	Porta vittoria	45.446205	9.239343	€ 6.450 /m²
4	5	Vigentino, Chiaravalle, Gratosoglio	29.87	123,779	4,487	Vigentino	45.419846	9.198153	€ 3.250 /m²
5	6	Barona, Lorenteggio	18.28	149,000	8,998	Barona	45.437929	9.145648	€ 3.200 /m²
6	7	Baggio, De Angeli, San Siro	31.34	170,814	6,093	Baggio	45.469598	9.114757	€ 2.400 /m²
7	9	Porta Garibaldi, Niguarda	21.12	181,598	9,204	Porta garibaldi	45.507704	9.179410	€ 6.800 /m²

Webscraping from Mercato Immobiliare to collect information into dataframe.

	Name	Price
0	Baggio	€ 2.400 /m²
1	Barona	€ 3.200 /m²
2	Centro storico	€ 7.100 /m²
3	Città studi	€ 4.500 /m²
4	Porta garibaldi	€ 6.800 /m²
5	Porta vittoria	€ 6.450 /m²
6	Stazione centrale	€ 5.000 /m²
7	Vigentino	€ 3.250 /m²

Mercato Immobiliare

Data Preparation - Immobiliare

1

```
In [13]: response2 = requests.get('https://www.mercato-immobiliare.info/lombardia/milano/milano.html').text
         soup2 = BeautifulSoup(response2, 'html.parser')
         table2 = soup2.find('table', {'id':'childrentable'})

In [14]: table_rows2 = table2.find_all('tr')

         res = []
         for tr in table_rows2:
             td = tr.find_all('td')
             row = [tr.text.strip() for tr in td if tr.text.strip()]
             if row:
                 res.append(row)


         df_price = pd.DataFrame(res, columns=["Name", "Price","Link"])
         df_price = df_price.drop(columns=['Link'])

In [15]: df_price['Name'] = df_price.Name.str.capitalize()
```

2

```
In [16]: df_price
```

Out[16]:

	Name	Price
0	Adriano	€ 2.750 /m²
1	Affori	€ 2.350 /m²
2	Baggio	€ 2.400 /m²
3	Barona	€ 3.200 /m²
4	Bicocca	€ 3.050 /m²
5	Bocconi	€ 5.300 /m²
6	Bovisa	€ 2.550 /m²
7	Bovisasca	€ 2.000 /m²
8	Brera	€ 7.600 /m²
9	Bruzzano	€ 2.100 /m²
10	Cadorna	€ 6.950 /m²

3

```
In [18]: df_top_townhall = pd.merge(df_top_townhall, df_price, on='Name', how='inner')

In [19]: df_top_townhall
```

Out[19]:

	Townhall	Description	Area	Population	Density	Name	Latitude	Longitude	Price
0	1	Centro storico	9.67	96,315	11,074	Centro storico	45.467281	9.185962	€ 7.100 /m²
1	2	Stazione Centrale, Gorla, Turro, Greco, Cresce...	12.58	153.109	13,031	Stazione centrale	45.499990	9.218999	€ 5.000 /m²
2	3	Città Studi, Lambrate, Porta Venezia	14.23	141,229	10,785	Città studi	45.489800	9.241031	€ 4.500 /m²
3	4	Porta Vittoria, Forlanini	20.95	156.369	8,069	Porta vittoria	45.446205	9.239343	€ 6.450 /m²
4	5	Vigentino, Chiaravalle, Gratosoglio	29.87	123,779	4,487	Vigentino	45.419846	9.198153	€ 3.250 /m²
5	6	Barona, Lorenteggio	18.28	149,000	8,998	Barona	45.437929	9.145648	€ 3.200 /m²
6	7	Baggio, De Angeli, San Siro	31.34	170,814	6,093	Baggio	45.469598	9.114757	€ 2.400 /m²
7	9	Porta Garibaldi, Niguarda	21.12	181,598	9,204	Porta garibaldi	45.507704	9.179410	€ 6.800 /m²

Point of interestes

	District	Dist_Latitude	Dist_Longitude	Venue	Venue_Lat	Venue_Long	Venue_Category
1	Centro storico	45.467281	9.185962	Starbucks Reserve Roastery	45.464920	9.186153	Coffee Shop
2	Centro storico	45.467281	9.185962	Ciacco. Gelato senz'altro	45.463704	9.186796	Ice Cream Shop
3	Centro storico	45.467281	9.185962	Gay Odin	45.466187	9.180801	Chocolate Shop
4	Centro storico	45.467281	9.185962	Di Viole Di Liquirizia	45.471460	9.185336	Cupcake Shop
5	Centro storico	45.467281	9.185962	Pasticceria Marchesi	45.465612	9.190091	Pastry Shop
6	Centro storico	45.467281	9.185962	Lavazza Coffee Design	45.466274	9.190975	Coffee Shop
7	Centro storico	45.467281	9.185962	Pasticceria Marchesi	45.468299	9.195018	Dessert Shop
8	Centro storico	45.467281	9.185962	Cova	45.468170	9.195347	Dessert Shop
9	Centro storico	45.467281	9.185962	Garibaldi Crème	45.474355	9.183466	Ice Cream Shop
10	Centro storico	45.467281	9.185962	Chocolat	45.467660	9.174160	Ice Cream Shop
11	Centro storico	45.467281	9.185962	Ammu - Cannoli espressi siciliani	45.466059	9.176548	Dessert Shop
12	Stazione centrale	45.499990	9.218999	Pasticceria Martesana	45.495824	9.203095	Dessert Shop
13	Stazione centrale	45.499990	9.218999	Gelateria Etnica	45.489586	9.217686	Ice Cream Shop
14	Stazione centrale	45.499990	9.218999	Pasticceria La Siciliana	45.488775	9.229817	Cupcake Shop
15	Stazione centrale	45.499990	9.218999	Gelateria Sartori	45.486618	9.206877	Ice Cream Shop
16	Stazione centrale	45.499990	9.218999	MAG - Mastri Artigiani del Gelato	45.481940	9.221057	Ice Cream Shop
17	Stazione centrale	45.499990	9.218999	Terra	45.482487	9.206623	Ice Cream Shop
18	Stazione centrale	45.499990	9.218999	Pasticceria Grossi	45.491280	9.237180	Dessert Shop
19	Stazione centrale	45.499990	9.218999	Pasticceria Ungaro	45.489943	9.237042	Dessert Shop
20	Stazione centrale	45.499990	9.218999	Pasticceria Valente	45.481174	9.220150	Dessert Shop

```
unique categories: 159
Italian Restaurant          78
Pizza Place                 52
Café                        31
Supermarket                 29
Hotel                       24
Ice Cream Shop              22
Park                        22
Restaurant                  20
Plaza                       17
Chinese Restaurant          17
Bakery                      17
Dessert Shop                16
Japanese Restaurant         13
Cocktail Bar                13
Pub                         12
Tram Station                11
Seafood Restaurant          11
Trattoria/Osteria           11
Sushi Restaurant            11
Art Gallery                 10
Gym                         10
Wine Bar                     9
Theater                      9
Soccer Field                 8
Boutique                     7
Asian Restaurant             6
Gym / Fitness Center         6
Nightclub                    6
Wine Shop                    6
Bar                          6
                            ..
```

By API we collect points of interests information from Foursquare , like venue, category and coordinates

FOURSQUARE

Data Preparation - Foursquare

1

```
In [20]: CLIENT_ID = 'TM3SILTIA0FQTOSSRL2RDH[redacted]BGKGV4FWYCQ2M'
         CLIENT_SECRET = 'UDGGMOMYUBWKY5G3I[redacted]CEVYBEMFQY5TUAW'
         VERSION = '20190325'
```

```
In [21]: radius = 2500
         LIMIT = 500

         def getNearbyVenues(names, latitudes, longitudes, radius=2500):

             venues_list=[]
             for name, lat, lng in zip(names, latitudes, longitudes):
                 print(name)

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

                 # make the GET request
                 results = requests.get(url).json()["response"]['groups'][0]['items']

                 # return only relevant information for each nearby venue
                 venues_list.append([(
                     name,
                     lat,
                     lng,
                     v['venue']['name'],
                     v['venue']['location']['lat'],
                     v['venue']['location']['lng'],
                     v['venue']['categories'][0]['name']) for v in results])

             nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
             nearby_venues.columns = ['District',
                           'Dist_Latitude',
                           'Dist_Longitude',
                           'Venue',
                           'Venue_Lat',
                           'Venue_Long',
                           'Venue_Category']

             return(nearby_venues)
```

```
In [22]: Milan_Venues = getNearbyVenues(names=df_top_townhall['Name'],
                                        latitudes=df_top_townhall['Latitude'],
                                        longitudes=df_top_townhall['Longitude']
                                       )

         Centro storico
         Stazione centrale
         Città studi
         Porta vittoria
         Vigentino
         Barona
         Baggio
         Porta garibaldi
```

2

```
In [23]: print ("Shape of the Venues Dataframe: ", Milan_Venues.shape)
         Milan_Venues.head(100)

         Shape of the Venues Dataframe:  (746, 7)
```

3

Out[23]:

	District	Dist_Latitude	Dist_Longitude	Venue	Venue_Lat	Venue_Long	Venue_Category
0	Centro storico	45.467281	9.185962	Starbucks Reserve Roastery	45.464920	9.186153	Coffee Shop
1	Centro storico	45.467281	9.185962	Gallerie d'Italia	45.467183	9.190056	Art Gallery
2	Centro storico	45.467281	9.185962	Galleria Vittorio Emanuele II	45.465577	9.190024	Monument / Landmark
3	Centro storico	45.467281	9.185962	Signorvino	45.467153	9.183460	Wine Bar
4	Centro storico	45.467281	9.185962	Teatro alla Scala	45.467027	9.189686	Opera House
5	Centro storico	45.467281	9.185962	Park Hyatt Milan	45.465532	9.188911	Hotel
6	Centro storico	45.467281	9.185962	Bulgari Lounge Bar	45.470014	9.188943	Cocktail Bar
7	Centro storico	45.467281	9.185962	Room Mate Giulia Hotel	45.465250	9.189396	Hotel
8	Centro storico	45.467281	9.185962	Giovanni Cova & C.	45.468816	9.184121	Bakery
9	Centro storico	45.467281	9.185962	Piazza Castello	45.468965	9.181312	Plaza
10	Centro storico	45.467281	9.185962	BVLGARI Hotel Milano	45.470149	9.189318	Hotel
11	Centro storico	45.467281	9.185962	Piazza del Carmine	45.470102	9.185058	Plaza
12	Centro storico	45.467281	9.185962	Piazza del Duomo	45.464190	9.189527	Plaza
13	Centro storico	45.467281	9.185962	Risoelatte	45.466514	9.183262	Italian Restaurant
14	Centro storico	45.467281	9.185962	Piazza della Scala	45.466895	9.189881	Plaza
15	Centro storico	45.467281	9.185962	Ciacco. Gelato senz'altro	45.463704	9.186796	Ice Cream Shop
16	Centro storico	45.467281	9.185962	Bialetti Store	45.464775	9.188343	Kitchen Supply Store
17	Centro storico	45.467281	9.185962	Moleskine Store	45.467531	9.183170	Paper / Office Supplies Store
18	Centro storico	45.467281	9.185962	Gay Odin	45.466187	9.180801	Chocolate Shop
19	Centro storico	45.467281	9.185962	Mandarin Oriental	45.469461	9.190876	Hotel
20	Centro storico	45.467281	9.185962	Castello Sforzesco	45.469545	9.180424	Castle
21	Centro storico	45.467281	9.185962	Di Viole Di Liquirizia	45.471460	9.185336	Cupcake Shop
22	Centro storico	45.467281	9.185962	Pasticceria Marchesi	45.465612	9.190091	Pastry Shop
23	Centro storico	45.467281	9.185962	Fontana del Castello Sforzesco	45.469237	9.180917	Fountain
24	Centro storico	45.467281	9.185962	Terrazze del Duomo	45.464207	9.191075	Scenic Lookout

Data Preparation - Foursquare

1

```
In [25]: # Create a Data-Frame out of it to Concentrate Only on sweet Place
Milan_sweet = Milan_Venues[Milan_Venues['Venue_Category'].str.contains('Coffee|Ice Cream|Chocolate|Cupcake|Pastry|Dessert|Pastry')].reset_index(drop=True)
Milan_sweet.index = np.arange(1, len(Milan_sweet)+1)
print ('Shape of the Data-Frame with Venue Category Cafe: ', Milan_sweet.shape)
Milan_sweet.head(100)

Shape of the Data-Frame with Venue Category Cafe:  (49, 7)
```

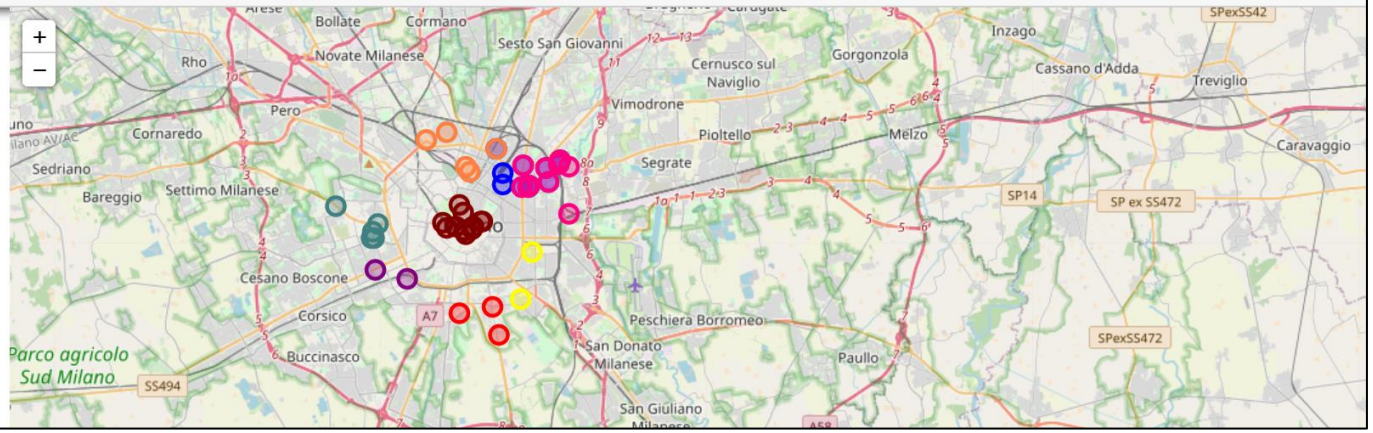
Out[25]:

	District	Dist_Latitude	Dist_Longitude	Venue	Venue_Lat	Venue_Long	Venue_Category
1	Centro storico	45.467281	9.185962	Starbucks Reserve Roastery	45.464920	9.186153	Coffee Shop
2	Centro storico	45.467281	9.185962	Ciacco. Gelato senz'altro	45.463704	9.186796	Ice Cream Shop
3	Centro storico	45.467281	9.185962	Gay Odin	45.466187	9.180801	Chocolate Shop
4	Centro storico	45.467281	9.185962	Di Viole Di Liquirizia	45.471460	9.185336	Cupcake Shop
5	Centro storico	45.467281	9.185962	Pasticceria Marchesi	45.465612	9.190091	Pastry Shop
6	Centro storico	45.467281	9.185962	Lavazza Coffee Design	45.466274	9.190975	Coffee Shop
7	Centro storico	45.467281	9.185962	Pasticceria Marchesi	45.468299	9.195018	Dessert Shop
8	Centro storico	45.467281	9.185962	Cova	45.468170	9.195347	Dessert Shop
9	Centro storico	45.467281	9.185962	Garibaldi Crème	45.474355	9.183466	Ice Cream Shop
10	Centro storico	45.467281	9.185962	Chocolat	45.467660	9.174160	Ice Cream Shop

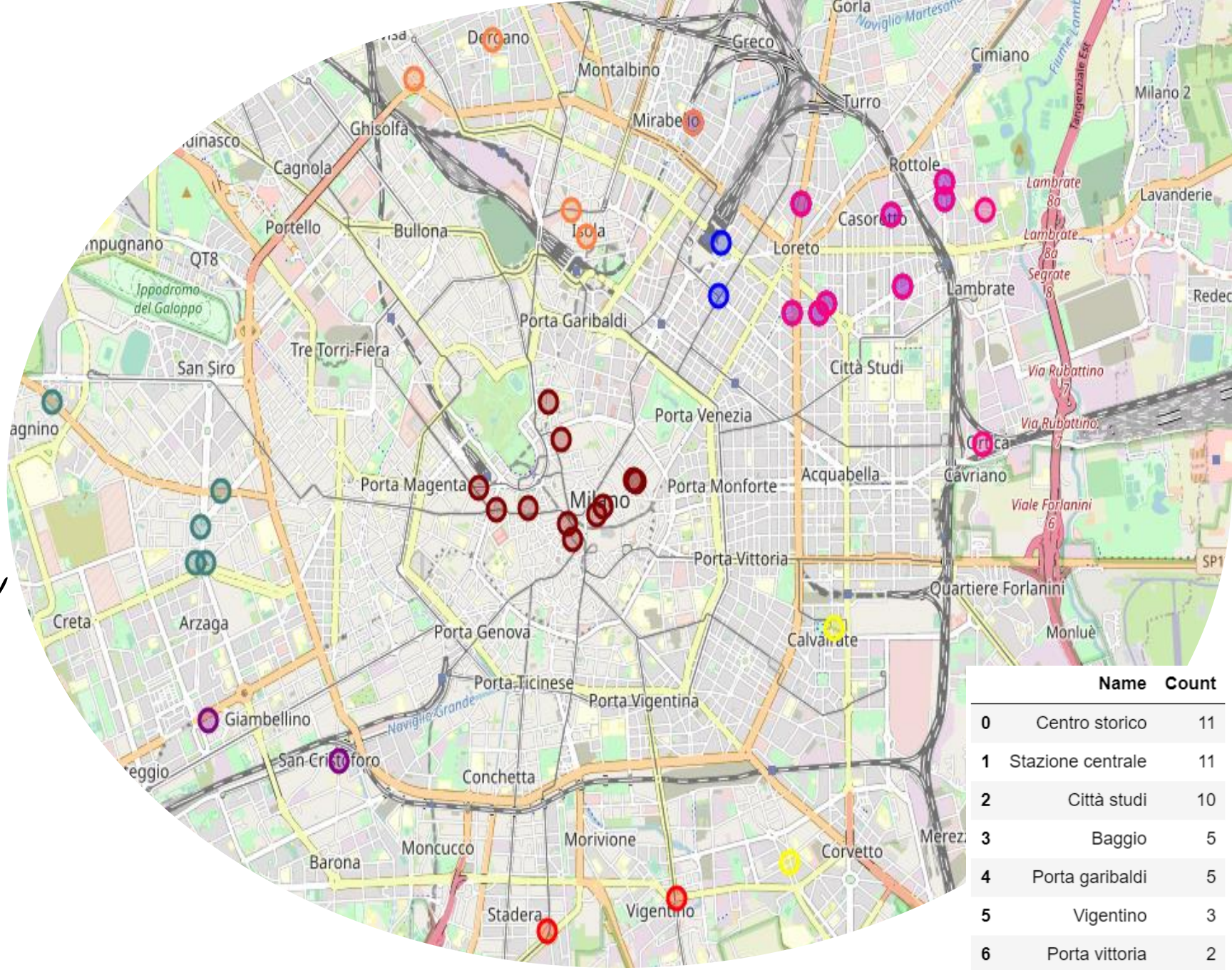
2

```
In [29]: map_sweet = folium.Map(location=[latitude, longitude], zoom_start=11, tiles="openstreetmap",
                               attr="<a href=https://github.com/python-visualization/folium/>Folium</a>")

Townhall = ["Centro storico", "Stazione centrale", "Città studi", "Porta garibaldi", "Porta vittoria", "Vigentino","Fiera","Barona","Baggio"]
x = np.arange(len(Townhall))
rainbow = ['#0000FF', '#FF0080','#FF8040','#FFFF00' ,'#FF0000', '#00FF00','#800080','#408080','#800000']

markers_colors = []
for lat, lon, poi, distr in zip(Milan_sweet['Venue_Lat'],
                                Milan_sweet['Venue_Long'],
                                Milan_sweet['Venue_Category'],
                                Milan_sweet['District']):
    label = folium.Popup(str(poi) + ' ' + str(distr), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=7,
        popup=label,
        color=rainbow[Townhall.index(distr)-1],
        fill=True,
        fill_color=rainbow[Townhall.index(distr)-1],
        fill_opacity=0.3).add_to(map_sweet)

map_sweet
```

Out[29]:



Data Visualization



	Name	Count
0	Centro storico	11
1	Stazione centrale	11
2	Città studi	10
3	Baggio	5
4	Porta garibaldi	5
5	Vigentino	3
6	Porta vittoria	2
7	Barona	2

Data Preparation - Foursquare

1

```
In [30]: # create a dataframe to calculate the number of cafes for each district
Milan_sweet_by_district = Milan_sweet['District'].value_counts()[0:10].to_frame(name='frequency')
Milan_sweet_by_district=Milan_sweet_by_district.reset_index()

Milan_sweet_by_district.rename(index=str, columns={"index": "Name", "frequency": "Count"}, inplace=True)
Milan_sweet_by_district
```

Out[30]:

	Name	Count
0	Centro storico	11
1	Stazione centrale	11
2	Città studi	10
3	Baggio	5
4	Porta garibaldi	5
5	Vigentino	3
6	Porta vittoria	2
7	Barona	2

2

```
In [31]: df_top_townhall = pd.merge(df_top_townhall, Milan_sweet_by_district, on='Name', how='inner')
```

```
In [32]: df_top_townhall
```

Out[32]:

	Townhall	Description	Area	Population	Density	Name	Latitude	Longitude	Price	Count
0	1	Centro storico	9.67	96,315	11,074	Centro storico	45.467281	9.185962	€ 7.100 /m²	11
1	2	Stazione Centrale, Gorla, Turro, Greco, Cresce...	12.58	153.109	13,031	Stazione centrale	45.499990	9.218999	€ 5.000 /m²	11
2	3	Città Studi, Lambrate, Porta Venezia	14.23	141,229	10,785	Città studi	45.489800	9.241031	€ 4.500 /m²	10
3	4	Porta Vittoria, Forlanini	20.95	156.369	8,069	Porta vittoria	45.446205	9.239343	€ 6.450 /m²	2
4	5	Vigentino, Chiaravalle, Gratosoglio	29.87	123,779	4,487	Vigentino	45.419846	9.198153	€ 3.250 /m²	3
5	6	Barona, Lorenteggio	18.28	149,000	8,998	Barona	45.437929	9.145648	€ 3.200 /m²	2
6	7	Baggio, De Angeli, San Siro	31.34	170,814	6,093	Baggio	45.469598	9.114757	€ 2.400 /m²	5
7	9	Porta Garibaldi, Niguarda	21.12	181,598	9,204	Porta garibaldi	45.507704	9.179410	€ 6.800 /m²	5

```
In [33]: df_top_townhall['Population'] = df_top_townhall['Population'].str.replace(',', '')
df_top_townhall['Population'] = df_top_townhall['Population'].str.replace('.', '')
df_top_townhall['Density'] = df_top_townhall['Density'].str.replace(',', '')
```

```
In [34]: df_top_townhall['Area'] = df_top_townhall['Area'].astype(float)
df_top_townhall['Population'] = df_top_townhall['Population'].astype(float)
df_top_townhall['Density'] = df_top_townhall['Density'].astype(float)

df_top_townhall
```

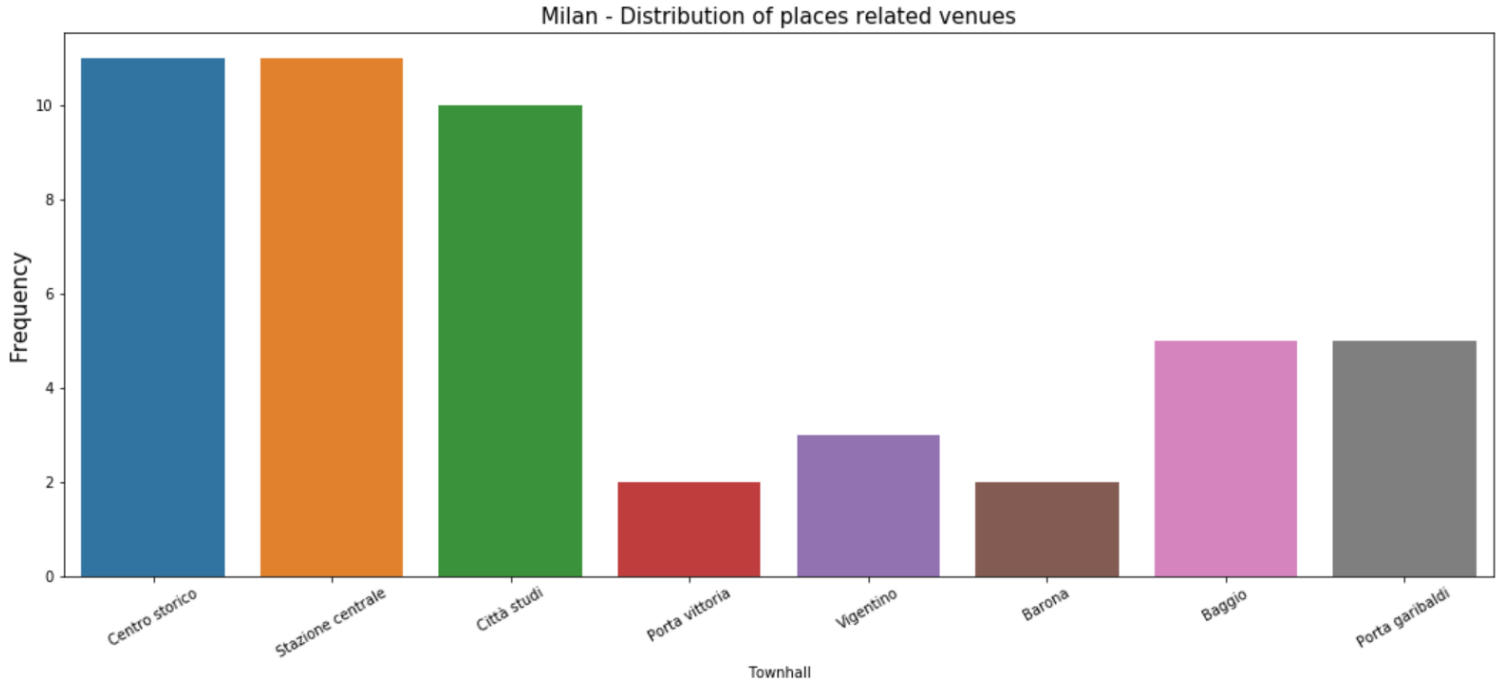
Out[34]:

	Townhall	Description	Area	Population	Density	Name	Latitude	Longitude	Price	Count
0	1	Centro storico	9.67	96315.0	11074.0	Centro storico	45.467281	9.185962	€ 7.100 /m²	11
1	2	Stazione Centrale, Gorla, Turro, Greco, Cresce...	12.58	153109.0	13031.0	Stazione centrale	45.499990	9.218999	€ 5.000 /m²	11
2	3	Città Studi, Lambrate, Porta Venezia	14.23	141229.0	10785.0	Città studi	45.489800	9.241031	€ 4.500 /m²	10
3	4	Porta Vittoria, Forlanini	20.95	156369.0	8069.0	Porta vittoria	45.446205	9.239343	€ 6.450 /m²	2
4	5	Vigentino, Chiaravalle, Gratosoglio	29.87	123779.0	4487.0	Vigentino	45.419846	9.198153	€ 3.250 /m²	3
5	6	Barona, Lorenteggio	18.28	149000.0	8998.0	Barona	45.437929	9.145648	€ 3.200 /m²	2
6	7	Baggio, De Angeli, San Siro	31.34	170814.0	6093.0	Baggio	45.469598	9.114757	€ 2.400 /m²	5
7	9	Porta Garibaldi, Niguarda	21.12	181598.0	9204.0	Porta garibaldi	45.507704	9.179410	€ 6.800 /m²	5

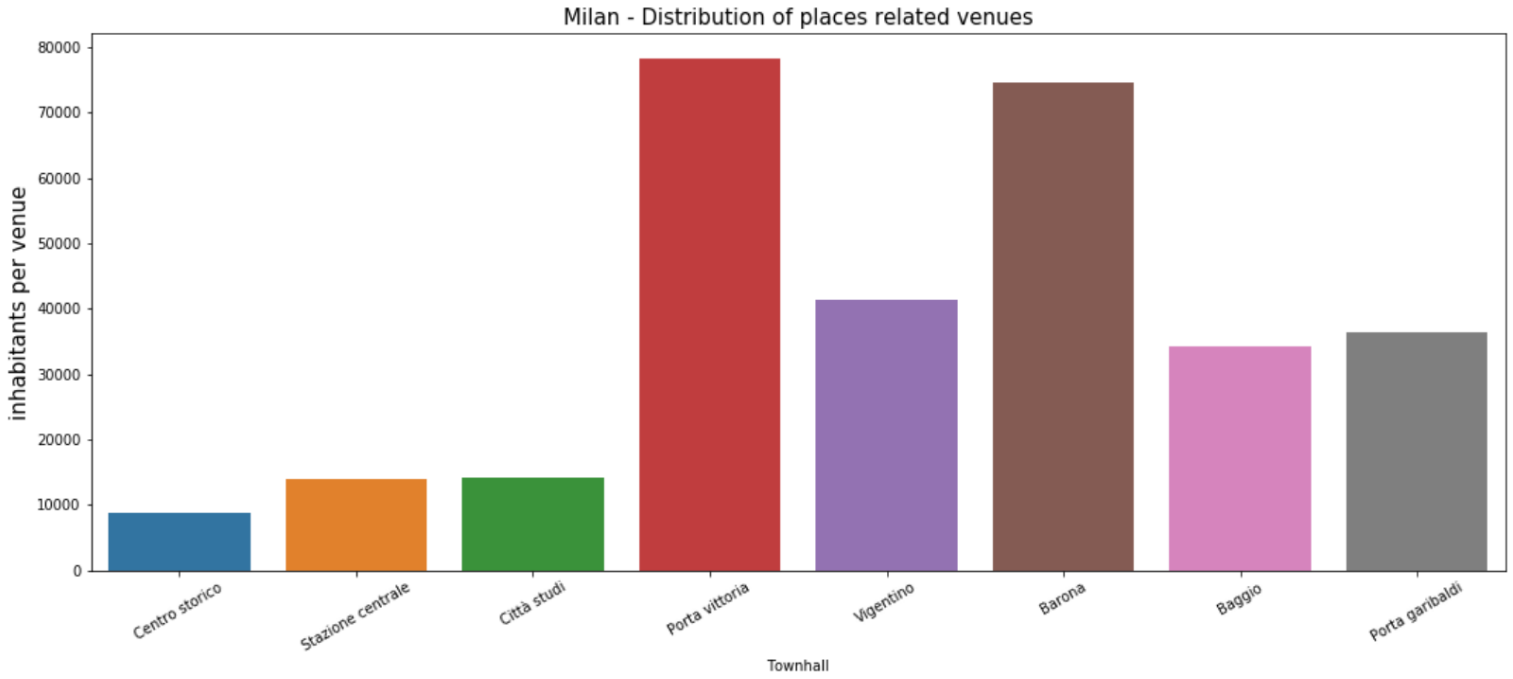
```
In [35]: df_top_townhall['Population By Venue'] = df_top_townhall['Population']/df_top_townhall['Count']
```

Final Dataset

	Townhall	Description	Area	Population	Density	Name	Latitude	Longitude	Price	Count	Population By Venue
0	1	Centro storico	9.67	96315.0	11074.0	Centro storico	45.467281	9.185962	€ 7.100 /m²	11	8755.909091
1	2	Stazione Centrale, Gorla, Turro, Greco, Cresce...	12.58	153109.0	13031.0	Stazione centrale	45.499990	9.218999	€ 5.000 /m²	11	13919.000000
2	3	Città Studi, Lambrate, Porta Venezia	14.23	141229.0	10785.0	Città studi	45.489800	9.241031	€ 4.500 /m²	10	14122.900000
3	4	Porta Vittoria, Forlanini	20.95	156369.0	8069.0	Porta vittoria	45.446205	9.239343	€ 6.450 /m²	2	78184.500000
4	5	Vigentino, Chiaravalle, Gratosoglio	29.87	123779.0	4487.0	Vigentino	45.419846	9.198153	€ 3.250 /m²	3	41259.666667
5	6	Barona, Lorenteggio	18.28	149000.0	8998.0	Barona	45.437929	9.145648	€ 3.200 /m²	2	74500.000000
6	7	Baggio, De Angeli, San Siro	31.34	170814.0	6093.0	Baggio	45.469598	9.114757	€ 2.400 /m²	5	34162.800000
7	9	Porta Garibaldi, Niguarda	21.12	181598.0	9204.0	Porta garibaldi	45.507704	9.179410	€ 6.800 /m²	5	36319.600000

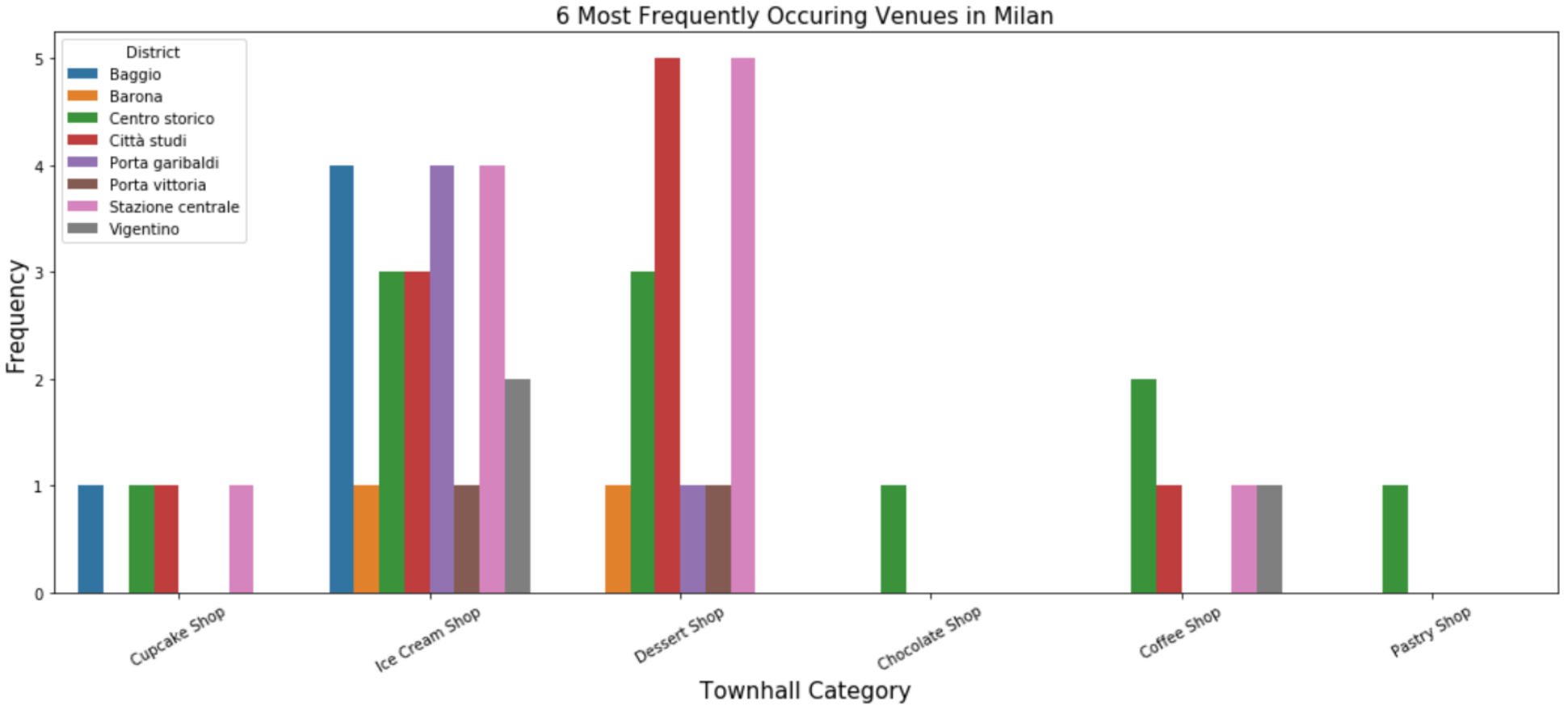


Distribution of Sweet places for each venues



Distribution of Sweet places for inhabitants

Distribution based category



	District	Venue_Category	count
0	Baggio	Cupcake Shop	1
1	Baggio	Ice Cream Shop	4
2	Barona	Dessert Shop	1
3	Barona	Ice Cream Shop	1
4	Centro storico	Chocolate Shop	1
5	Centro storico	Coffee Shop	2
6	Centro storico	Cupcake Shop	1
7	Centro storico	Dessert Shop	3
8	Centro storico	Ice Cream Shop	3
9	Centro storico	Pastry Shop	1
10	Città studi	Coffee Shop	1
11	Città studi	Cupcake Shop	1
12	Città studi	Dessert Shop	5
13	Città studi	Ice Cream Shop	3
14	Porta garibaldi	Dessert Shop	1
15	Porta garibaldi	Ice Cream Shop	4
16	Porta vittoria	Dessert Shop	1
17	Porta vittoria	Ice Cream Shop	1
18	Stazione centrale	Coffee Shop	1
19	Stazione centrale	Cupcake Shop	1
20	Stazione centrale	Dessert Shop	5
21	Stazione centrale	Ice Cream Shop	4
22	Vigentino	Coffee Shop	1
23	Vigentino	Ice Cream Shop	2

Final results & open points

- Analysis shows that the number of places (Cupcake, Ice Cream, Coffe, Chocolate, Dessert, Pastry) is quite low considering the population in Milan.
- Dessert & Ice Cream shops are the best category in Milan; these shops are concentrated in Stazione Centrale & Città Studi (North area of Milan).
- Porta Vittoria and Barona are the districts with the highest ratio inhabitants per townhall/venue (over than 70.000). Average real estate price shows Barona is cheaper than Porta Vittoria for 50% of the price.
- Porta Vittoria (Est Area) and Barona (South Area) are the districts with a small number of places in target.