Milan’s Neighborhoods
Analysis and Ranking

Edoardo Olivieri
Federica Romano
Francesca Verna

Table of Contents

Brief
Introduction

This project aims to build a comprehensive database to help users identify the best **neighborhoods** in Milan based on their specific needs.

The study focuses on users' **priorities**, offering insights into Milan’s urban landscape and supporting informed relocation decisions.

Data Acquisition

Overpass Turbo

Defining neighborhood borders and locating supermarkets

Yelp API

Retrieving information on restaurants, museums, and nightlife locations

Downloaded Datasets

Retrieving remaining information from City of Milan Open Data Portal and Osservatorio del Mercato Immobiliare

OverpassTurbo
Query

Associating points of interest with neighborhoods was a challenge.
Official datasets **lacked** predefined neighborhood **boundaries**, and many amenities retrieved through the API listed broad locations to bypass the Yelp algorithm and ensure visibility.

To solve this, custom neighborhood **polygons** were created using Overpass Turbo, a tool for querying OpenStreetMap.
These polygons were saved in GeoJSON format and used for accurate spatial mapping, ensuring that data points were assigned to the correct neighborhoods.
Additionally, a specific **query** on Overpass Turbo was used to retrieve data on supermarkets in each neighborhood.

YELP
API

The Yelp API was used to gather data on **restaurants**, **museums**, and **nightlife**.
Data **requests** were made using Python's requests library, using as inputs the neighborhood names from the GeoJSON file, obtained through Overpass Turbo queries.

With authentication via a **private API key**, up to 240 business results were retrieved per neighborhood. Key details like business name, address, ratings, and coordinates were collected. To mitigate potential issues, **error handling** was implemented to log and skip neighborhoods where HTTP errors occurred. Furthermore, a brief **delay** was introduced between requests to adhere to API rate limits and ensure uninterrupted data retrieval.

Finally, the data was stored in a **Pandas DataFrame**, converted into a GeoDataFrame for spatial analysis.

Datasets
Download

City of Milan: Open Data Portal
These GeoJSON datasets included information on schools, parks, pharmacies, transport stops, and other key locations. Each dataset was loaded into a GeoDataFrame using the geopandas library and converted to a common Coordinate Reference System.

Osservatorio del Mercato Immobiliare
To gather real estate data, both tabular and geospatial datasets were obtained. The tabular dataset includes property and rental values in euros per square meter, along with property types. The GeoJSON file provides geospatial data on real estate zoning and boundaries.

A directory for storing all downloaded GeoJSON files was created to ensure proper organization, and error handling was added to ensure reliable data retrieval.

Data Preparation
Cleaning and Integration

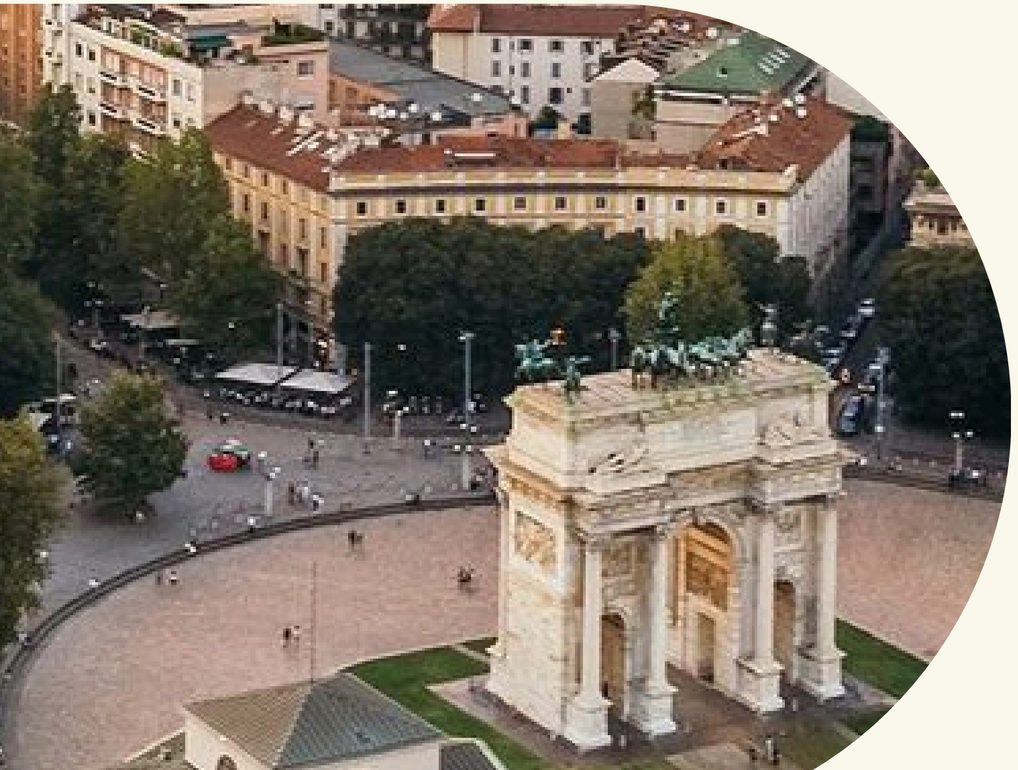
Data Cleaning:

It is crucial to evaluate the quality of the data by checking for missing values, duplicates, errors, or inconsistencies.

Data Integration:

Spatial Joins were conducted to integrate the various datasets with the GeoJSON file containing the neighborhood boundaries.

Data Cleaning



We performed data cleaning by **removing rows** with missing/invalid geometries or neighborhood values and **ensuring** that all datasets had **valid** geospatial matches for analysis.

Since each dataset was different, there was **no universal**, one-size-fits-all **approach** to further clean the data, therefore, each dataset needed its own considerations.

Restaurants, Museums, Nightlife	Removed duplicates based on name, address, categories, ratings, and geometry
Dog Parks and Playgrounds	Retained duplicates across neighborhoods to reflect shared spaces
Sports Venues	Removed generic entries when specific labels existed. Standardized activity labels. Excluded irrelevant entries.
Universities	Removed duplicates while preserving unique entries by faculty and location.
Schools	Combined public and private school datasets, retaining all unique programs. Added a column to distinguish private and public schools.
Transportation Systems	Standardized columns across train, metro, and bus datasets. Merged into a single GeoDataFrame.

Data Cleaning
Missing Values

While most datasets were complete and ensured high data quality, a few exceptions required careful handling:

- *Restaurants Dataset*: out of 2597 entries, 1031 lacked the Price variable, 6 the Categories variable and 1 the Business Address variable. These entries were retained since they still provided valuable insights like ratings and reviews. Imputation was avoided to prevent introducing bias.
- *Housing Prices Dataset*: Missing data for the neighborhoods were addressed by imputing global averages for essential pricing variables, ensuring consistent comparisons across neighborhoods.
- *Coworking Spaces Dataset*: four entries lacked Opening Hours. No imputation was performed due to the lack of reliable estimates.
- *Supermarkets Dataset*: four records had missing Names, but these were retained to maintain a complete view of supermarket presence across neighborhoods.
- *Nightlife Dataset*: 2 missing observations for the Categories variable.
- *Museums Dataset*: 1 missing entry for the Categories variable.

Data Integration
Spatial Join

To integrate the data, a **spatial join** was performed between the datasets and the GeoJSON file containing Milan's neighborhood boundaries. Using the sjoin function with *how=left* we ensured that all primary records were retained, while *predicate=within* matched each data point to its respective neighborhood.

This operation was key to **associating business** locations **and venues** with specific neighborhoods, enabling analysis and comparison of amenities across different areas of Milan.

Data Integration
Home Prices

For real estate price data, **preprocessing** was required before integration. The dataset was filtered for 2024 records and specific property categories, such as Abitazioni civili and Ville e Villini. An average purchase price column was calculated from the minimum and maximum values.

A **left outer join** linked this data to geospatial real estate zones via the Zona variable. Next, a **spatial join** with *predicate=intersects* was performed to match overlapping geometries, associating property prices with Milan's neighborhoods. **Aggregated metrics**, including minimum, maximum, and average prices, were then calculated at the neighborhood level while preserving their spatial boundaries for analysis.

Data Storage

To efficiently handle and analyze complex geospatial data, **MongoDB** was chosen for its flexible, schema-less structure and native support for GeoJSON and geospatial queries. This made it well-suited for **integrating diverse datasets**, including points of interest (POIs), housing prices, and transportation networks, enabling a comprehensive evaluation of Milan's neighborhoods.

We created the *ranking_milano* database, which contains a collection of neighborhood documents, composed of key fields such as geometry, a nested locations dictionary for categorizing POIs, and a home_prices dictionary for property price statistics.

Finally, the **structured documents** were inserted into MongoDB, an extensible database optimized for geospatial and categorical queries.

Queries

1.Most Diverse Neighborhoods

2.Demographic-Based Neighborhood Scoring

3.Neighborhood Matching Based on User Preferences

Our aim is to determine the best neighborhood based on different needs and preferences.

Query
Most Diverse Neighborhoods

This query identified the top five Milan neighborhoods with the **most diverse amenities,** by calculating a diversity score based on the number of distinct categories available.

Rank	Neighborhood	Diversity score
1	Guastalla	13
2	Bovisa	13
3	Buenos Aires-Venezia	13
4	Città Studi	12
5	Bicocca	12

Neighborhoods were ranked in descending order, with the top five selected for their broad range of services.

This analysis is particularly valuable for individuals looking for areas that offer a **rich variety of services** to meet different needs and lifestyles.

Query
Demographic-Based Neighborhood Scoring

This query systematically evaluates Milan’s neighborhoods based on their **suitability** for **students, families, and singles/couples**.

The scoring process considers amenities and housing prices, **normalizing and weighting** categories based on their relevance to each demographic.
Housing prices are factored in by applying a penalty to expensive areas to balance affordability.

Neighborhoods are ranked in descending order, with scores scaled into percentages for clarity. The top five for each category are highlighted, identifying the most suitable areas for different lifestyles.

Single/Couples

Rank	Neighborhood	Score (%)
1	Buenos Aires - Venezia	100.00
2	Duomo	71.11
3	Città Studi	57.33
4	Niguarda - Cà Granda	54.22
5	Villapizzone	52.31

Computed by prioritizing transport, nightlife, coworking spaces, and sports venues, rather than schools or playgrounds.

Families

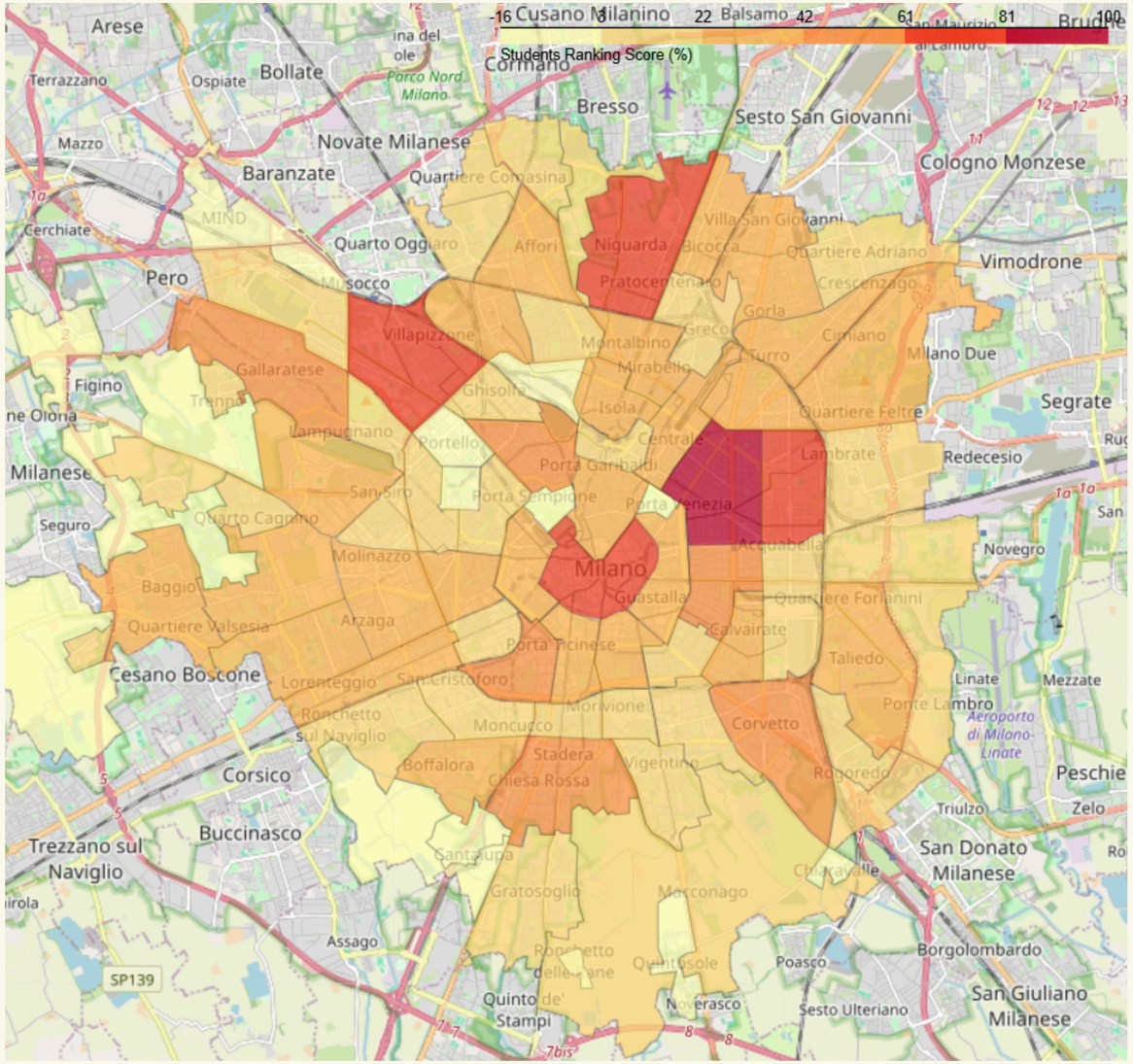
Rank	Neighborhood	Score (%)
1	Niguarda - Cà Granda	100.00
2	Buenos Aires - Venezia	98.89
3	Villapizzone	81.92
4	Stadera	81.13
5	Gallaratese	72.32

Computed by prioritizing schools, playgrounds , dog parks, rather than night life or universities.

Students

Rank	Neighborhood	Score (%)
1	Buenos Aires - Venezia	100.00
2	Città Studi	69.46
3	Niguarda - Cà Granda	65.09
4	Duomo	64.24
5	Villapizzone	64.00

Computed by prioritizing prices, nightlife, universities, and sports venues, rather than schools, restaurants or playgrounds.



Query
Neighborhood Matching Based on User Preferences

This query is designed to gather **user preferences interactively** and then find the three neighborhoods that best match those preferences.

The result is a list of neighborhoods that best meet the user's needs, with details about the match score, average price, and a breakdown of which requirements were met and which were not. If any of the user-specified conditions are not fulfilled by a neighborhood, they are flagged accordingly.

To illustrate how the query works, we created an example based on a *physics student* who has just transferred to Milan.

Rank	Neighborhood	Not Fulfilled Categories	Score
1	Città Studi	Budget	10
2	Stadera	Faculty, Restaurant Cat.	9
3	Quarto Cagnino	Faculty, Library	9

Conclusion
and Future Developments

By analyzing public amenities, housing prices, and neighborhood characteristics, the project provided **valuable insights** for individuals assessing which neighborhoods best suit their needs. The rankings and diversity scores offer a **solid foundation for people considering a move to Milan**.

Future Developments:

1. Incorporating Crime Data;
2. Improving Demographic Scoring;
3. Refining Weight Values;
4. Public Surveys and Expert Input;
5. Scaling for Other Cities;
6. Enhanced Personalized Suggestions.

The End