DATA AND INFORMATION QUALITY PROJECT REPORT

PROJECT ID: 28

DATASET: 8

Lei Leonardo C.P. 10710185

Mantegazza Niccolò C.P. 10710172

A.A 2024/2025

POLITECNICO

MILANO 1863

1. Setup choices

Regarding the choice of libraries, we selected those introduced during the exercise sessions. Specifically, we used ydata-profiling for most of the data profiling tasks and recordlinkage for data deduplication. We did not use any machine learning-based libraries because the dataset was highly descriptive, and these libraries did not provide additional useful insights.

For data preparation techniques, we followed the most relevant steps covered in the lectures.

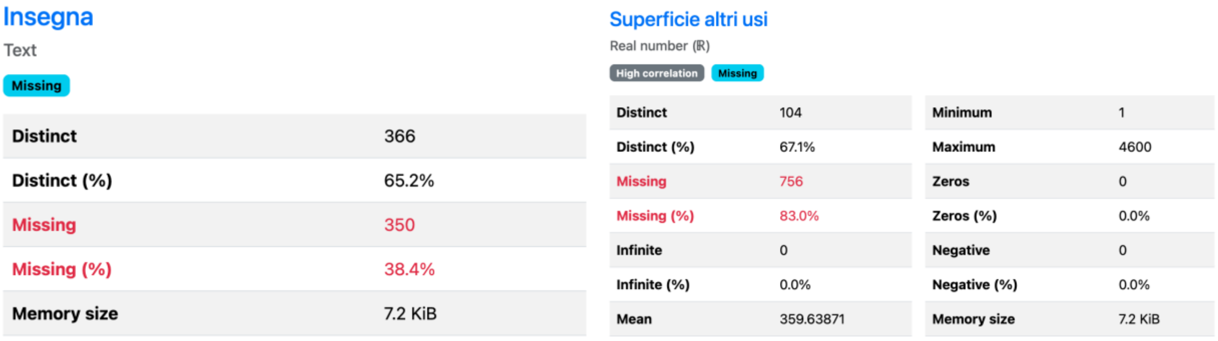
2. Pipeline implementation

Data profiling

To generate the data profile, we used the ProfileReport function from the ydata_profiling library. During this step, we identified several issues with the dataset.

First, some columns had a significant number of missing values (e.g., the 'Insegna' attribute had 38.4% missing data, while 'Superficie altri usi' had as much as 83.0%). Additionally, some columns contained messy or inconsistent value names, such as 'Settore merceologico'.

This exploration step also allowed us to study the distribution of attribute values using histograms, where we identified potential outliers in certain columns.



Insegna

Text

Missing

Distinct	366
Distinct (%)	65.2%
Missing	350
Missing (%)	38.4%
Memory size	7.2 KiB

Superficie altri usi

Real number (ℝ)

High correlation Missing

Distinct	104	Minimum	1
Distinct (%)	67.1%	Maximum	4600
Missing	756	Zeros	0
Missing (%)	83.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	359.63871	Memory size	7.2 KiB

Figura 1 - Insight of some attributes

Another consideration involves the correlation between 'ZD' and 'Codice Via', which is approximately 0.62. An even higher correlation was observed between 'Superficie Vendita' and 'Superficie Totale', which was expected, as we know the 'Superficie' attributes are highly correlated. In fact, one is likely the sum of the other two.

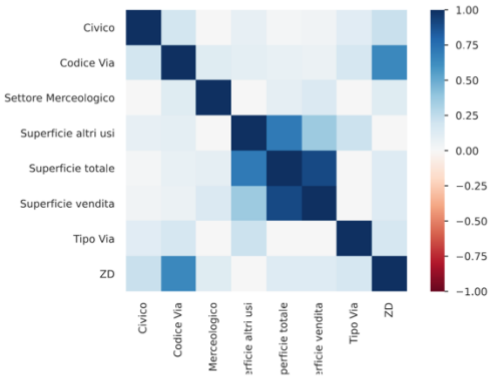


Figura 2 – Pearson Correlation

Initial quality assestment

As part of this step, we computed various metrics to evaluate data quality dimensions for each column. The results confirmed what was observed during the profiling step: several columns lack completeness. Additionally, we gained insights into the uniqueness and distinctiveness of the values. Some columns exhibited very high uniqueness levels, which could make it challenging to predict missing values effectively.

Functional dependency and integrity constraints

Using domain knowledge, we identified that the same address (a combination of 'Tipo Via' and 'Via') should correspond to the same 'Codice Via' and 'ZD'. While the first rule was consistently respected, the second was not.

Furthermore, we know that the sum of 'Superficie Vendita' and 'Superficie altri usi' must equal 'Superficie Totale'. We flagged these integrity constraints and corrected the tuples that violated them to ensure consistency.

Data transformation/Standardization

Schema normalization

We observed redundancy in the location data, as the same information was stored in the 'Ubicazione' attribute, as well as in multiple attributes like 'Tipo via', 'Via', 'Civico', and 'ZD'. To achieve a more efficient structure, we decided to keep the individual, original split attributes, as they provide a clearer and more organized foundation. Consequently, we dropped the 'Ubicazione' column.

For consistency, we kept the 'copy columns' extracted from the 'Ubicazione' data, which will be useful in the error detection phase. These copy columns have the same name as the original ones, but in CAPS LOCK to distinguish them.

As for the additional data, such as "accesso", we initially extracted it but found that it contained a total of null values and unintelligible entries of 104. Given its lack of relevance to the dataset, we decided to drop this column. Similarly, the 'isolato' attribute was deemed unnecessary, as the remaining information was sufficient to define the location of the store or activity, so we dropped it along with the original 'Ubicazione' column.

We opted to keep the location information split across multiple attributes, such as 'Tipo via', 'Via', and 'Codice Via', to enhance readability and make future analysis easier.

Attribute renaming

To improve clarity and consistency, we renamed 'Via' to 'Denominazione' to align with this Comune di Milano document and to avoid confusion with 'Tipo Via'. We also renamed 'ZD' to 'Municipio', as this is the local term used in Milan. Additionally, we added the appropriate units of measurement to the relevant columns. (*Note: The 'Superficie' attributes make use of '_' because of necessity to avoid syntax error in python. The '_' will be removed at end after the cleaning pipeline*)

Data type conversions and values standardization

We standardized the data types to appropriate formats:

- 'Civico' and the newly created 'CIVICO' column were converted from float64 and object to Int type, as these values are clearly integers and need to be comparable.

- We also standardized the values in the ‘Settore merceologico’ column to reduce inconsistencies and improve data quality.

Data cleaning

Error detection

Consistency Check for ‘Ubicazione’ data with Via, Tipo Via, Zona:

Our base rule was that the original dataset’s ‘Tipo Via’ column had no missing values and contained valid labels. We used this as the foundation and prioritized it over the extracted values from ‘Ubicazione’.

By comparing the two columns, ‘TIPO VIA’ and ‘Tipo Via’, we identified rows where the values differed. To check for errors, we verified if both ‘Tipo Via’ and ‘Denominazione’ were correct, as a street’s full address should always be consistent when referring to the same location. The discrepancies were observed only when the entire address was different, and we applied our predefined policy for handling such cases.

Since the functional dependency (‘Tipo Via’, ‘Denominazione’) → (‘Zona’) was violated, we flagged these violations and corrected them by using the most frequent (mode) value.

Outlier detection & correction

To detect outliers in ‘Codice Via’, we used the Z-score with a threshold of 3.

When examining these outliers, we compared them to the values extracted from ‘Ubicazione’. We found no significant differences between the two versions, so we decided to retain the outliers, as we had no concrete evidence to determine whether they were valid or invalid.

We also checked if using the mode to ensure consistency for the dependency (‘Tipo Via’, ‘Denominazione’) → (‘Zona’) was a suitable solution by searching if any of the violating rows were outliers and none were, so we evaluated it as decent enough.

We also dropped the duplicate columns (‘TIPO VIA’, ‘DENOMINAZIONE’, ‘ZONA’) as they were no longer useful for further analysis.

For the ‘Superficie Vendita’ and ‘Superficie Totale’ columns, we considered values of (0, 1) as outliers, as these are unrealistic for the specific domain they represent. However, for ‘Superficie Altri Usi’, we assumed that some businesses may not have any surface area used for other purposes. So, we treated values of (0, 1) as invalid (set to Nan) and corrected them during the missing value imputation step.

Missing values imputation

Settore merceologico missing values

This attribute had a small percentage of missing values (1.3%), so we decided to use a simple imputation technique while considering the potential bias this might introduce. We implemented an imputation algorithm based on domain knowledge of popular supermarkets in Milan. Specifically, if the ‘Insegna’ attribute matched a popular supermarket name, we imputed the value as ‘alimentare/non alimentare’. If there was no match, we imputed the value with the mode, as it was the most frequent.

Superficie vendita/altri usi/totale missing values

After invalidating outliers in these columns, many new missing values were created, adding to those that were already present. For imputation, we exploited the integrity rules we defined earlier and computed the missing column values based on the other two columns. When this wasn’t possible due to two or more missing attributes, we handled the cases as follows:

- If exactly two columns were missing and one of them was ‘Superficie altri usi’, we imputed It as 0 and computed the missing one based on the non-missing column. This approach was valid, because, as we understood in the profiling step, in many cases a missing value on ‘Superficie altri usi’ simply meant that it was 0.
- If exactly two columns were missing and ‘Superficie altri usi’ was not one of them, or all three columns were missing, we simply decided to drop the row since it wasn’t possible to have an estimate of those value. Luckily, no tuples fell into this scenario so it was not a problem at all.

After imputing the missing values, we rechecked the integrity constraints for the ‘Superficie Totale’ attribute and found that some tuples were still in violation. We addressed this issue by recalculating the attribute values of ‘Superficie totale’ for the violating tuples, ensuring the integrity constraints were met. Finally, we identified one tuple that violated a constraint where ‘Superficie Vendita’ exceeded ‘Superficie Totale’. We corrected this by adjusting the ‘Superficie Vendita’ value to align with the integrity rule.

Insegna missing Value choice:

We noticed that ‘Insegna’ had a significant number of missing values (up to approximately 1/3 of the data), making it impractical to drop the rows with missing values. Since ‘Insegna’ is an object-type attribute, statistical or ML-based imputation methods weren’t applicable. One option was to drop the column altogether, but we decided against this to retain the valuable information. Instead, we filled the missing values with “Mancante” to preserve the data.

Civico missing Value

We observed that ‘Civico’ had 72 missing values. However, the corresponding values extracted from the ‘Ubicazione’ column had no missing values, though 13 of them were invalid (‘CIVICO’ = 0). A quick comparison revealed 108 rows of difference between the two columns, which was more than the 72 missing values in ‘Civico’. Therefore, we decided to keep the values from ‘Civico’ where they differed and imputed the missing values using ‘CIVICO’, in line with our original policy. Since ‘Civico’ was specifically created for this purpose, we deemed it to be more accurate than the messy ‘Ubicazione’ column. After this, only 4 invalid addresses remained, so we dropped these 4 rows, considering it acceptable given the dataset’s size (over 900 rows). We preferred this approach over using standard imputation techniques, which had a low probability of being accurate.

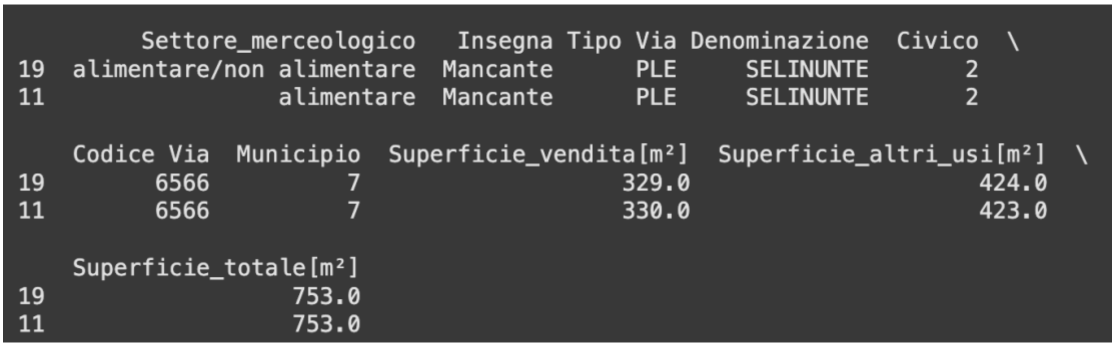
Data deduplication

As first step, we dropped the exact duplicates from the dataset. To detect non-exact duplicates, we imported the recordlinkage library and used the Jaro-Winkler distance function to compute address similarity.

Our approach for duplicate detection assumed that an identical address has a very high probability of representing the same tuple.

Therefore, we performed exact matching on ‘Municipio’, ‘Codice Via’, and ‘Civico’, and used the Jaro-Winkler distance (with a threshold of 0.8) to compare the ‘Tipo Via’ and ‘Denominazione’ attributes.

Finally, we retained only one instance of each set of potential duplicate entries, resulting in the removal of many (redundant) rows.

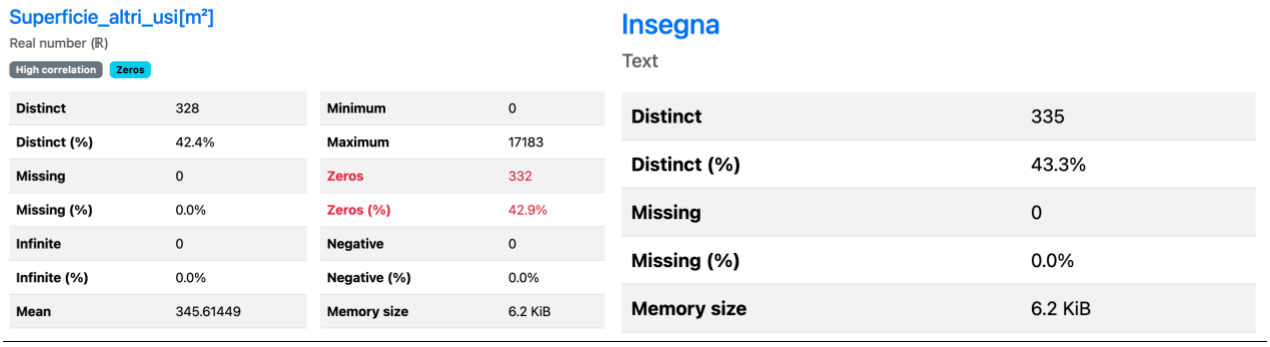


```
        Settore_merceologico   Insegna Tipo Via Denominazione  Civico  \
19  alimentare/non alimentare  Mancante      PLE     SELINUNTE       2
11                 alimentare  Mancante      PLE     SELINUNTE       2

    Codice Via  Municipio  Superficie_vendita[m²]  Superficie_altri_usi[m²]  \
19        6566          7                   329.0                     424.0
11        6566          7                   330.0                     423.0

    Superficie_totale[m²]
19                  753.0
11                  753.0
```

Figura 3 - Potential duplicates has the same exact address

Final quality assessment and conclusion

After completing all the cleaning steps, we performed a final round of profiling and quality assessment to compare various aspects of the data before and after cleaning. As a result, the completeness of the database is now 100%, and the uniqueness of some columns has improved due to the elimination of duplicates. In addition, all the integrity constraints are now satisfied. Although the dataset size has decreased from 911 rows to 773, the remaining tuples now contain more values and offer more useful information.



Superficie_altri_usi[m²]

Real number (ℝ)

High correlation Zeros

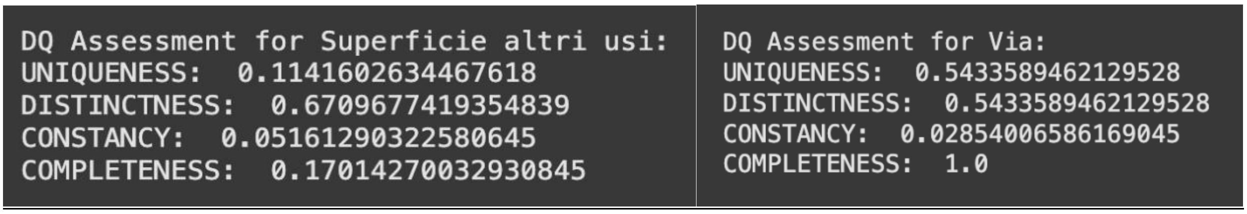
Distinct	328	Minimum	0
Distinct (%)	42.4%	Maximum	17183
Missing	0	Zeros	332
Missing (%)	0.0%	Zeros (%)	42.9%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	345.61449	Memory size	6.2 KiB

Insegna

Text

Distinct	335
Distinct (%)	43.3%
Missing	0
Missing (%)	0.0%
Memory size	6.2 KiB

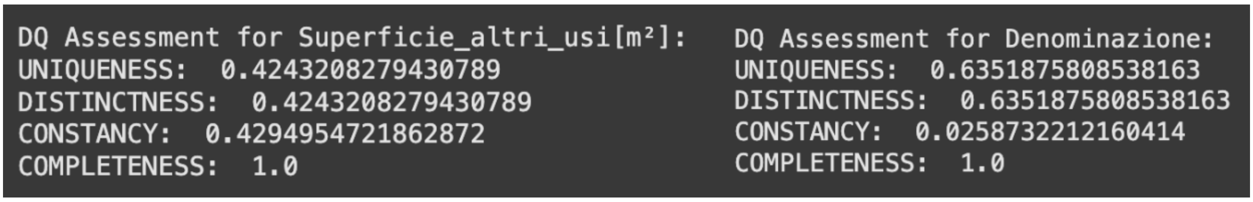
Figura 4 - Insight of the same attribute as the beginning



```
DQ Assessment for Superficie altri usi:
UNIQUENESS:  0.1141602634467618
DISTINCTNESS:  0.6709677419354839
CONSTANCY:  0.05161290322580645
COMPLETENESS:  0.17014270032930845
```

```
DQ Assessment for Via:
UNIQUENESS:  0.5433589462129528
DISTINCTNESS:  0.5433589462129528
CONSTANCY:  0.02854006586169045
COMPLETENESS:  1.0
```

Figura 5 – Some DQ before



```
DQ Assessment for Superficie_altri_usi[m²]:
UNIQUENESS:  0.4243208279430789
DISTINCTNESS:  0.4243208279430789
CONSTANCY:  0.4294954721862872
COMPLETENESS:  1.0
```

```
DQ Assessment for Denominazione:
UNIQUENESS:  0.6351875808538163
DISTINCTNESS:  0.6351875808538163
CONSTANCY:  0.0258732212160414
COMPLETENESS:  1.0
```

Figura 6 - Some DQ after