POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Data and Information Quality: Data Preparation Pipeline

Author(s): **Marta Monsó (11074367)**
Miguel Planas (11071870)
(Erasmus+ students)

Group ID: **31**

Assigned Dataset: **6**

Academic Year: 2024-2025

Contents

1 | Set up choices

1.1. Libraries and data preparation techniques

To ensure the successful execution of the project, several libraries and techniques were employed to establish a robust data preparation pipeline. Below is a detailed account of the setup choices:

- **Libraries:**
  - `pandas`: For data manipulation and cleaning, including operations like handling missing values, detecting duplicates, and performing transformations.
  - `numpy`: Used for numerical computations and handling missing data efficiently.
  - `datetime`: To standardize and manipulate date-related columns.
  - `seaborn` and `matplotlib`: For visualizing data distributions and identifying potential outliers.
  - `ydata_profiling`: For generating an in-depth profiling report of the dataset, helping to identify key issues such as missing values, outliers, and data types.
  - `efficient_apriori`: Reserved for potential association rule analysis (if relevant).
- **Data Loading and Preprocessing:**
  - The dataset was loaded using `pandas`, and an initial inspection was performed to assess its structure. This included verifying the shape of the dataset, column names, and data types.
  - Column names were standardized by converting them to lowercase and replacing spaces with underscores to maintain consistency.
  - Missing values in critical columns such as `Permit Number`, `Permit Type`, and `Street Name` were quantified to evaluate their impact on the dataset.

Dataset Overview: The dataset, *Building Permit Applications Data*, provides detailed records of building permit applications from San Francisco between January 1, 2013, and February 25, 2018. It includes approximately 200,000 records and 43 attributes, covering information such as `Permit Number`, `Permit Type`, `Street Name`, and `Issue Date`. This data is crucial for analyzing the building permit process, which is a key factor in the real estate industry's supply and demand dynamics. Understanding patterns in permit issuance can shed light on potential delays and their implications for urban development.

2 | Pipeline Implementation

2.1. Data exploration and profiling

The initial step of the project involved a thorough exploration and profiling of the dataset ‘building_permits.csv‘. This process aimed to understand the structure, contents, and quality of the data to identify potential issues and prepare for subsequent cleaning and transformation steps.

2.1.1. Dataset Overview

The dataset contained the following key attributes:

- **Permit Number**: Unique identifier for each permit.
- **Permit Type**: Category of the permit.
- **Location**: Geographic coordinates of the permit (latitude and longitude).
- **Dates**: Including Permit Creation Date, Issued Date, and Expiration Date.
- **Cost Information**: Estimated and Revised costs of the project.

2.1.2. Profiling Metrics

The profiling was conducted using Python libraries such as `pandas` and `ydata-profiling`, generating a comprehensive summary of the dataset. Key profiling outputs included:

- **Dataset Shape**: The dataset contained `198900` rows and `43` columns.
- **Data Types**: A breakdown of data types across columns.
- **Missing Values**: Identified columns with missing data and calculated the percentage of missing entries.
- **Value Distributions**: Visualized histograms for numerical columns and value counts for categorical columns, as seen in 2.1.

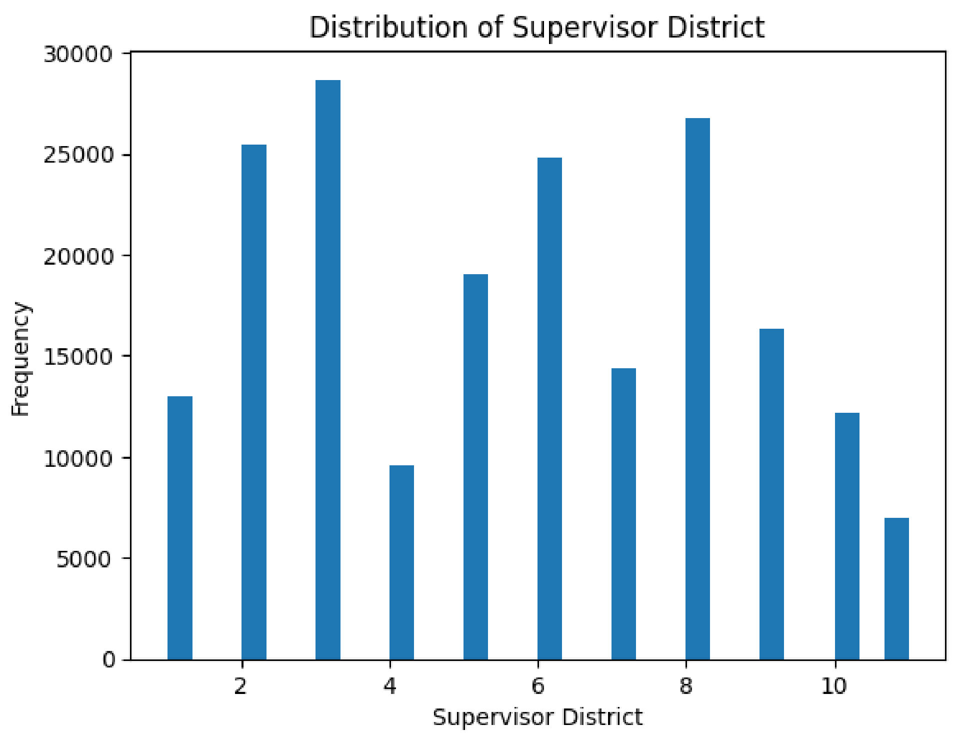

Figure 2.1: Example of the histograms to analyze value distributions.

2.1.3. Observations

During the profiling process, the following issues were identified:

- High levels of missing data in several columns, such as `Street Number Suffix`.
- Duplicate entries in key columns, including `Permit Number`.
- Presence of outliers in numerical columns, such as Estimated Cost and Revised Cost.
- Inconsistent formatting of categorical values, like varying capitalization in `Permit Type`.

This step provided a comprehensive understanding of the dataset's quality by generating metadata, which includes insights on data types, distributions, and missing values. Data profiling is an essential precursor to data cleaning and transformation, as it ensures that subsequent processes are well-informed and targeted. By uncovering structural issues, logical inconsistencies, and data anomalies, profiling enables the development of a cleaning pipeline tailored to address the dataset's unique challenges.

2.2. Data Quality Assessment

The second step focused on assessing the quality of the dataset by evaluating key dimensions of data quality: completeness, accuracy, consistency, and validity. This assessment was guided by established data quality frameworks to ensure a systematic and comprehensive analysis.

2.2.1. Completeness

The completeness of the dataset was evaluated by calculating the total number of missing values and their distribution across columns. Columns with high percentages of missing values, such as `Street Number Suffix` and `Voluntary Soft-Story Retrofit`, were identified as critical areas requiring targeted imputation or logical substitutions.

2.2.2. Accuracy

Numerical columns, such as `Estimated Cost` and `Revised Cost`, were examined for negative or illogical values. Furthermore, categorical columns were checked for invalid entries against predefined domain constraints, ensuring that all data values adhered to expected formats and ranges.

2.2.3. Consistency

Data consistency was verified by detecting duplicate records and ensuring that unique identifiers, such as `Permit Number`, did not have multiple occurrences. Additionally, relationships between related columns were checked for logical coherence, such as ensuring that `Issued Date` did not precede `Permit Creation Date`.

2.2.4. Validity

Validity checks focused on ensuring that all date fields adhered to logical timelines and that numerical columns fell within realistic ranges. Outliers were identified using statistical methods, including the Interquartile Range (IQR).

2.3. Data cleaning

After the data exploration and profiling phases, we moved on to Data Cleaning to perform all the necessary transformations on the dataset. This step is essential to improve its quality and make it more optimal for subsequent analysis. In this phase, we focused on data transformation and standardization. We addressed missing values using various techniques to correct them, analyzed outliers and finally, removed rows with potential duplicate data.

2.3.1. Data Transformation/Standardization

First, we began by analyzing the values in each category using the results obtained from the previous phases of the project.

To simplify this step, we started by removing columns with more than 80 percent missing values, as they did not provide relevant information. As with such a high proportion of missing data, the remaining values would not accurately represent the underlying distribution or offer meaningful insights. The columns removed were: `Street Number Suffix`, `Unit Suffix`, `Structural Notification`, `Voluntary Soft-Story Retrofit`, `Site Permit`, `Fire Only Permit`, `Unit` and `TIDF Compliance`.

We performed several data operations to improve the dataset's structure and usability:

1. **Combining Columns**: We merged the columns `Street Suffix`, `Street Name`, and `Street Number` into a single column called `Street Address` to create a unified category.

2. **Splitting Columns**: The `Location` column was split into two separate columns named `Latitude` and `Longitude`, allowing for easier manipulation of these variables individually.

3. **Formatting Corrections**:

   - All columns representing dates were converted to the `datetime` format.
   - We verified that all numerical categories were correctly formatted; since no inconsistencies were found, no changes were made.
   - For categorical variables, all text was converted to lowercase to ensure uniform formatting.

4. **Simplifying Categories**: The column `Proposed Use` contained two categories: `1 Family Dwelling` and `2 Family Dwelling`. These were combined into a single category called `Family Dwelling`, as the distinction between the two did not provide significant additional information.

5. **Removing Redundant Columns**: The columns `Existing Construction Type`

and `Proposed Construction Type` contained the same information as `Existing Construction Type Description` and `Proposed Construction Type Description`, but without the additional detail provided in the latter columns. To reduce redundancy, we removed the less informative columns.

6. **Analyzing Relationships Between Related Columns**:

   We analyzed the differences between `Existing Construction Type Description` and `Proposed Construction Type Description` and found that 151876 values were identical. Similar comparisons were performed for:

   - `Number of Existing Stories` and `Number of Proposed Stories`.
   - `Estimated Cost` and `Revised Cost`.
   - `Existing Use`, `Proposed Use`, `Existing Units`, and `Proposed Units`.

7. **Creating New Columns**:

   - `Cost Difference`: This new column was created to capture the difference between `Estimated Cost` and `Revised Cost`. It can be used in future analyses to assess whether estimated prices are consistent and logical or require improvement.
   - `On Time Completion`: A binary column was added to indicate whether permits were completed before their scheduled end date. This allows for quick analysis of project timeliness.

8. **Reorganizing Columns**: To improve the dataset's readability after the modifications, we adjusted the column order, ensuring that related information was grouped logically.

These transformations ensured a cleaner, more organized dataset, making it easier to perform further analysis and derive meaningful insights.

2.3.2. Error detection and correction of missing values

Managing missing data was a crucial step in our data preprocessing process, as it directly impacted the quality and reliability of subsequent analysis and modeling. Below, we provide a detailed explanation of the steps we took to handle missing values in our dataset.

We began by calculating the total number of missing values and the corresponding percentage for each column in the dataset. Identifying the distribution of missing data across columns helped us decide how to handle these values. Calculating percentages provided a clear, normalized view of the prevalence of missing values.

Steps Taken to Correct Missing Values

1. **Handling "Completed Date" and "Issued Date":**

   - We observed that it was normal for the variable `Completed Date` to have missing values (NaN) if the current status for the building was not marked as "complete." For each building that was not completed, we replaced the NaN values with the text *"Status not completed."*
   - Similarly, for the `Issued Date` column, missing values were likely due to permits not being issued for those buildings. We replaced these NaN values with the text *"Status not issued."*

2. **Using KNNImputer for Numerical Columns:**

   - We employed the `KNNImputer` from `sklearn` to impute missing values based on the nearest neighbors. This method was applied to numerical columns, including *"Number of Existing Stories," "Existing Units," "Estimated Cost,"* and *"Plansets."* KNNImputer considers the local structure of the data by imputing values based on the nearest neighbors, thereby preserving patterns and variability within the dataset that might otherwise be lost with simpler methods. This ensured better imputations for these columns.

3. **Interpolating "Revised Cost" and Calculating "Cost Difference":**

   - Missing values in the `Revised Cost` column were interpolated based on the `Estimated Cost` column using linear interpolation.
   - Additionally, we created a new column called *"Cost Difference"* to capture the difference between `Estimated Cost` and `Revised Cost`. For any missing values in `Cost Difference`, we calculated and filled them with the difference between the two related columns.

4. **Imputing "Description" Column:**

   - Missing values in the `Description` column were replaced with the text: *"No description associated with the type of building."* This approach provided a clear message about the absence of data, which was preferable to leaving the values empty or filling them with unrelated information, as it would have been difficult to determine a concrete description for missing values.

5. **Predicting Latitude and Longitude with Linear Regression:**
   - We used `Linear Regression` to predict missing `Latitude` and `Longitude` values based on the `Street Number` column.
   - This method was chosen because of the logical linear relationship between street numbers and geographical coordinates. The regression model leveraged this sequential progression to estimate coordinates, ensuring that the imputed values aligned with the geographic trends in the dataset.
6. **Predicting Zipcode and Supervisor District:**
   - Once missing values for `Latitude` and `Longitude` were imputed, we used `Linear Regression` again to predict missing values in the `Zipcode` and `Supervisor District` columns based on the `Latitude` and `Longitude` variables.
7. **Replacing Neighborhood Missing Values:**
   - For the `Neighborhoods - Analysis Boundaries` column, we replaced missing values with the most frequent category. This decision was based on the assumption that the missing values likely belonged to the most common neighborhood.
8. **Median Imputation for Proposed Features:**
   - Missing values in the numerical columns *"Number of Proposed Stories"* and *"Proposed Units"* were imputed using the median.
   - Since most of the values in these columns were clustered around specific values, imputing missing data with the median made sense as it represented the central tendency of the data.
9. **Using IterativeImputer for Categorical Columns:**
   - To handle missing values in the following categorical columns: *"Existing Use," "Proposed Use," "Existing Construction Type Description,"* and *"Proposed Construction Type Description,"* we used the `IterativeImputer`.
   - This method modeled each feature with missing values as a function of the other features and iterated over the data until convergence, leveraging relationships between features to generate plausible imputations.

After implementing these steps, we successfully obtained a dataset with no missing values, ensuring that the data was clean, consistent, and ready for further analysis.

2.3.3. Localization and correction of inconsistencies (outliers)

Outliers were identified and corrected to ensure that the dataset reflected accurate and meaningful values. Various methods were employed to detect and address these anomalies, as outlined below:

1. **Detection of Outliers:**

   - The Interquartile Range (IQR) method was applied to numeric columns to identify outliers. Any value outside the range of $Q1 - 1.5 \times IQR$ and $Q3 + 1.5 \times IQR$ was flagged as an outlier.

   - Boxplots were generated for visual inspection of the data distributions and to confirm the presence of outliers.

2. **Handling Detected Outliers:**

   - For columns such as `Number of Existing Stories`, `Number of Proposed Stories`, `Existing Units`, and `Proposed Units`, outliers were replaced with the median value of their respective columns to maintain the central tendency of the data.

   - A similar approach was applied to other numeric variables with significant outliers, ensuring that the adjusted values did not distort the dataset's overall characteristics.

3. **Advanced Detection Techniques:**

   - The **Isolation Forest algorithm** was used to identify potential outliers in high-dimensional data. This unsupervised learning method detected anomalies based on data points' isolation levels in the feature space.

   - Columns such as `Latitude`, `Longitude`, and `Zipcode`, which inherently contain edge values, were reviewed contextually to determine whether flagged outliers were valid or needed adjustments.

4. **Validation of Corrections:** After the corrections, the distributions of the adjusted columns were re-evaluated to ensure consistency with expected patterns and to confirm that the imputed values were reasonable. Additionally, visualizations such as histograms and boxplots were re-generated to verify the absence of extreme outliers, ensuring the cleaned dataset maintained its integrity and reliability.

The detection and correction of outliers significantly improved the dataset's reliability and usability, ensuring that analyses based on this data would yield credible and accurate insights.

2.3.4. Data Deduplication

Data deduplication was performed to enhance the consistency and reliability of the dataset by identifying and removing redundant records. The deduplication process consisted of the following steps:

Detection and Removal of Exact Duplicates

Initially, exact duplicates were identified and removed using the `drop_duplicates()` method:

- Rows with identical values across all columns were flagged as duplicates.
- Duplicate rows were dropped, reducing the dataset size to `(181495, 35)`.

This ensured that only unique records remained in the dataset.

Detection of Near-Duplicates

To address potential near-duplicates, the `recordlinkage` library was utilized:

- A **Sorted Neighborhood Index** was created using `Permit Number` as the primary key with a sliding window size of 9 to generate candidate links.
- Candidate pairs were compared using a combination of exact and similarity-based metrics:
  - **Exact Matching:** Columns such as `Permit Number` were required to match exactly.
  - **Similarity-Based Matching:** Attributes like `Street Address`, `Neighborhoods - Analysis Boundaries`, `Permit Type`, and `Description` were compared using metrics like Jaro-Winkler similarity with a threshold of 0.85.

Resolution of Matches

Matches with high similarity scores (above a specified threshold) were resolved by retaining the most complete or recent record.

Final Dataset

The deduplication process achieved the following results:

- **Redundant Rows Removed:** All duplicate and near-duplicate records were successfully eliminated, resulting in a refined dataset with `181,495` unique records.
- **Output File:** The deduplicated dataset was saved as `DATA_DROP_DUPLICATES.csv`, ensuring accessibility for further analysis.

This systematic approach to deduplication enhanced the dataset's overall quality by removing redundant information and preserving only meaningful, unique entries. As a result, the dataset is now more reliable, consistent, and ready for downstream analysis or modeling tasks.

2.4. Step 4: Final Quality Assessment and Project Conclusion

The final step involved assessing the quality of the cleaned and transformed dataset to ensure that all quality dimensions were satisfactorily addressed. This step also highlights the importance of a robust data preparation pipeline for achieving high data quality.

2.4.1. Final Quality Assessment

A systematic evaluation of the dataset was conducted across the following dimensions:

- **Completeness:** All missing values were either imputed or logically handled, reducing the overall missing data percentage to near-zero levels.
- **Accuracy:** Numerical columns were validated to ensure no negative or illogical values, and categorical columns adhered to predefined domain constraints.
- **Consistency:** Duplicate records were completely removed, and unique identifiers such as `Permit Number` were verified to ensure uniqueness.
- **Timeliness:** Date fields were checked for logical relationships, such as ensuring `Issued Date` occurred before `Permit Expiration Date`.

3 | Project Conclusion

This project underscored the critical importance of a robust data preparation pipeline in achieving high-quality datasets. Key takeaways include:

- **Value of Data Profiling:** Profiling provides essential metadata and insights that guide targeted cleaning and transformation efforts.
- **Impact of Data Cleaning:** Addressing missing values, removing duplicates, and resolving inconsistencies ensures data reliability and usability.
- **Foundation for Analysis:** High-quality data is foundational for any downstream analysis, as it minimizes biases and enhances the credibility of results.

In conclusion, a well-executed data preparation pipeline is a cornerstone of data quality management. By systematically addressing quality dimensions such as completeness, accuracy, and consistency, the project achieved a clean and reliable dataset ready for further analysis and decision-making.