DigitMan

Occupant-based DIGITal predictive MANteinance to improve the built environment

Grant number: 2022JMRX2A

WP 2 – Safety predictive module development

T.2.1 – Relevant maintenance KPIs definition
T.2.2 – Maintenance data acquisition and post-processing
T.2.3 – Maintenance method development

DELIVERABLE ID	D2.1
Deliverable Title	KPIs and predictive methods for maintenance tasks
Date	28/02/2025
Last revision date	25/02/2025
Revision	002
Main partner	UNIVPM
Additional partners	
Authors of the contribution	Marco D'Orazio, Guido Romano
Deliverable type	Report
Number of pages	15

Abstract

This deliverable describes the action performed to develop the module addressed to predict the priority to assign to end-users' maintenance requests collected the two ticketing support systems in use at UNIVPM and POLIMI. A unified database has been created to align UNIVPM and POLIMI tickets and to integrate structured and unstructured data to support decision-making processes. Following a rigorous text pre-processing phase, NLP methods, including normalisation, stop-word removal, stemming, lemmatization, and regex data cleaning, have been applied to optimize the dataset. Key Performance Indicators (KPIs) were defined to assess building health status, maintenance efficiency, and request prioritization. The most relevant KPIs are also selected for future multi-criteria evaluations and improve comparisons between different areas and activities. A Bidirectional Long Short-Term Memory (Bi-LSTM) model was implemented to automatically predict request priorities, demonstrating high accuracy (0.87) and reliable classification metrics. This predictive approach enables efficient allocation of maintenance resources and can support decision-makers in feature prediction, allowing real-time automated classification with human supervision. The integration of KPIs and ML-based classification methods within a digital dashboard facilitates a comprehensive assessment of maintenance needs, improving response times and resource optimization. Future developments within WP5 will focus on integrating the selected KPIs into assessment metrics, weighting and merging them with additional "how-to" microservices. The integration of these methodologies within a BIM-based and spatially structured framework will further enhance facility management by linking current and predictive maintenance insights directly to specific building stock elements.

Keywords

Building Maintenance, Key performance indicators, predictive module, machine learning, data mining, text mining

Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA

Finanziato dall'Unione europea NextGenerationEU

Ministero dell'Università e della Ricerca

UNIVERSITÀ POLITECNICA DELLE MARCHE

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

POLITECNICO MILANO 1863

Approvals

Role	Name	Partner	Date
Coordinator	Marco D'Orazio	UNIVPM	28/02/2025
Task leader	Marco D'Orazio	UNIVPM	28/02/2025

Revision versions

Revision	Date	Short summary of modifications	Partner
001	30/01/2025	Revision of terminology and accuracy results update	UNIVPM
001	25/02/2025	Text formatting and grammar checking	UNIVPM

Summary

Introduction

Building facility managers should face different challenges to support timely and proper maintenance tasks in building stocks of large organisations, in case corrective interventions are needed [1, 2]. In particular, the most relevant ones concern the analysis of the building health status to detect “hot-spots” for maintenance requests and deploy best related management strategies, and (2) the reduction of timings in interventions along with the increase in the intervention reliability.

Decision-making in building maintenance could take advantage of data-driven approaches based on large databases about maintenance interventions, end-user requests and reports from technicians related to occurred failures [1, 3, 4]. These data are generally collected thanks to Building Automation Systems (BAS) and Computerized Maintenance and Management Systems (CMMS) [3], which are relevant data sources since they can connect facility managers with occupants too. In fact, CMMS can store communication from occupants about detected failures implying corrective maintenance interventions, and their organization usually involves many features describing the whole management process, starting from the end-user request and ending with the technician’s intervention validation. Stored data can include textual information as well as numerical data (including time-related ones) and other features divided into categories (e.g. typology of issue, position, priority, status of the request) [5].

Data-driven approaches to building maintenance [3] use data analytics and machine learning to supply facility managers with informed decisions, thus contributing to reduce downtime, extend the lifespan of assets, and improve the quality of life for occupants. Although different approaches have been provided to this end by previous works, also including the use of textual (thus unstructured) data from end-users (through end-user communication) [2–10], it is worth noting that relevant lacks in the comprehensive and effective definition of simple key performance indicators to drive facility managers’ decision seem to exist, indeed. Machine learning (ML) techniques have been also widely applied, but their application into rapid operational tools for decision makers seem to be still limited, especially considering multiple purposes related to automatic prediction and given the large contemporary number of requests that can appear in large organizations [5, 9]. In this context, ones of the most relevant issues relate to dataset pre-processing actions, information alignment and categorization, selection of ML in the pillar of natural language processing of unstructured end-users’ textual requests, and correlation between automatic assignment tasks by ML and decision makers’ supervised choices.

Providing tools to support decision makers in quick analysis of critical issues for current scenarios can boost the assessment process in the “how-to” process to check the current “health” status of the building, mainly using a user-centered approach linked to satisfaction of end-users and workers (including technicians) [3, 10, 11]. In this sense, the connection of maintenance data-driven outcomes with building modelling approaches could be relevant to ensure a multi-purpose approach with other facilities management pillars, as those related to energy efficiency, occupancy optimization, and safety management, which can take advantage of multicriteria analysis [3].

In view of the above, this report aims at:

- Defining Key Performance Indicators (KPIs) for maintenance-related analysis in “how-to” scenarios, which can be used to assess the building health status,
- Defining predictive methods to rapidly support decision makers in the automatic assignment of categories for each end-user request,
- and defining the process of input data collection and calculation to derive the KPIs and to implement the predictive methods, thus moving towards the deployment of digital tools combined with CMMSs.

1 Phases and methods

According to the work aims, the current work is organized into two main phases. The first phase concerns the KPIs definition (Section 1.1), while the second one concerns the data acquisition and post-processing, and thus involves the related calculation method development and tool development (Section 1.2).

1.1 Criteria for Key Performance Indicators Definition

KPIs are defined according to the "how-to" perspectives for maintenance tasks defined in WP1. From a general perspective, KPIs should follow the SMART assessment approach [20], being:

- **S**pecific: Targeting a specific maintenance issue for the building, comprising all the elements included in the building stocks, according to consolidated classification rules (i.e. OMNICLASS-based);
- **M**easurable: Quantifying maintenance needs of the building, while being comparable in terms of output range. For instance, normalization of KPIs should be encouraged to make them vary between maximum and minimum maintenance needs and efforts conditions, or describing the fulfilment of requirements and end-users' expectation in the same interval;
- **A**ssignable: Assigning to one or more element of the building, or, at least, at areas or buildings within the specific building stock of the large organization, so as to supply decision makers with information about the impact of building current scenarios (in "how-to" perspectives);
- **R**ealistic: Establishing objectives related to maintenance optimization in "how-to" conditions, mainly relying on end-users' requests as fundamental data in large organizations, while being also supervised by decision-makers, to address any additional constraint from a general operational level;
- **T**ime related: Focusing on quick and timely analysis of maintenance conditions, and supporting decision makers and maintenance staff in rapid assessment of maintenance requests (e.g. by typology, severity, position), to avoid procrastination and reducing time efforts for manual data analysis which is still widely performed in maintenance flows.

The same criteria are applied to (1) the analysis of the current status of building health, thanks to data mining techniques on collected data, and to (2) the definition of predictive methods to automatize the recognition and classification of basic features from end-users textual requests (i.e. priority, programmability, typology of interventions), using ML techniques.

In view of the above, KPIs have been defined to rapidly evaluate and quantify multiple but disconnected pieces of information supplied by the intervention request databases. KPIs enable the systematic analysis and interpretation of data by standardizing the way in which requests are categorized and assessed. For this reason, they also needed an accurate classification of maintenance interventions into homogeneous categories (Types) and sub-categories to ensure the collection of specific tasks in the different pilot features. Therefore, a preliminary activity to align the maintenance interventions typology starting from the pilot facility management contracts has been provided using the general logics in Section 5.1. The OmniClass structure (derived from Table 22) was selected as common framework and adapted depending on specificities of the application context in terms of operational and technological issues, to unequivocally define the DigitMan Activity type list, according to WP1 activities.

Concerning the analysis of the current status of building health, KPIs have been first defined by comprising a specific unit of measure. Refinements of KPIs have been provided to reduce their number in order to provide basic maintenance assessment through simple but reliable indicators. Then, these fundamental KPIs have also been modified to make them range between 0 (minimum impact of maintenance needs) and 1 (maximum impact). Percentage terms have been also considered within the same rationale. This allows to establish basis for WP5 multicriterial analysis, obtaining and managing comparable KPIs. In both cases, KPIs are also defined to provide specific value by end-user request or by request classifications (e.g. building area, intervention typology), and their percentile-based analysis (i.e. using 5th, 50th and 95th percentiles as relevant values).

Concerning the definition of predictive methods to automatize the recognition and classification of basic features from end-users textual requests, KPIs correspond to priority, programmability, and typology of interventions, and their related labels, which prediction has been performed using ML techniques previously validated in other maintenance contexts [5].

Although they are derived for the Italian context and specifically for educational buildings, due to DigitMan application scenario, the KPIs are applicable also in other contexts of large organization which essentially adopt the same parameters to described maintenance needs.

1.2 Data acquisition and methods implementation

1.2.1 Overview

Data acquisition and methods take into account the development of the OMNICLASS-based structure for activities and space classification provided by WP1, and then rely on the following activities:

1. Definition of input data structure to organize a structured database, aligning features of end-users requests from the considered application context (POLIMI and UNIVPM), as shown in Section 1.2.2;
2. Selection of tools and methods for "database pre-processing" to calculate the KPIs, as shown in Section 1.2.3;
3. Definition of methods for the implementation for the analysis of the current status of building health, thanks to data mining techniques on collected data, as shown in Section 1.2.4;
4. Definition of predictive methods to automatize the recognition and classification of basic features from end-users textual requests (i.e. priority, programmability, typology of interventions), using ML techniques, as shown in Section 1.2.5.
5. Multicriteria analysis (to be addressed in WP5).

The final application is provided on the whole POLIMI and UNIVPM building stocks introduced in WP1 activities, by considering differences among the

1.2.2 Database structuring

The database structure has been provided starting from the input features of the single databases on maintenance requests of the POLIMI and UNIVPM building stocks introduced in WP1. Basic criteria include: (1) removal of non-useful information for the KPIs, including anonymization of requests; (2) connection of common fields from the databases, aggregation of multiple information, removal or integration of non common fields; (3) uniform standard definition of classes of values, especially for those related to labels; (4) implementation of calculation steps to aggregate numerical data when needed.

Table 1 provides an overview of the alignment of UNIVPM and POLIMI pilots databases about the fields selected in the following steps. The alignment of location data ("*isPointOf*") considers different spatial organizations of the real estate assets and the structure of the reporting systems of the two Universities. At UNIVPM, the available data allows localization only at the building level, whereas at POLIMI, it is possible to achieve a more detailed resolution down to single rooms. Moreover, at UNIVPM different districts (e.g., Engineering, Agricolture) may be housed within the same building cluster, which in turn can be displaced across different addresses within the same city (e.g., Ancona).

Table 1. Alignment of databases from UNIVPM and POLIMI pilots.

DigitMan	Typology	Description	UNIVPM	POLIMI
UID	*KEY*	*Maintenance request identifier*	*ID*	*Protocollo*
pr_Description	*VARCHAR(255)*	*Request description in text format*	*Descrizione chiamata*	*Descrizione*

DigitMan	Typology	Description	UNIVPM	POLIMI
rl_IsPointOf_BuildingStock	*VARCHAR(255)*	*University (Univpm or Polimi)*	-	-
rl_IsPointOf_District	*VARCHAR(255)*	*District*	*Building_group*	*Luogo*
rl_IsPointOf_Site	*VARCHAR(255)*	*Address*	*Indirizzo*	*Città*
rl_IsPointOf_BuildingCluster	*VARCHAR(255)*	*District code*	*Code*	*Comprensorio*
rl_IsPointOf_Building	*VARCHAR(255)*	*Building*	*Building*	*Edificio*
rl_IsPointOf_BuildingStorey	*VARCHAR(255)*	*Floor*	-	*Piano*
rl_IsPointOf_Space	*VARCHAR(255)*	*Room*	-	*Vano*
rl_Initiates_Activity	*VARCHAR(255)*	*Relationship between request and activity*	*ID Ticket*	*Protocollo OTRS*
pr_Priority	*VARCHAR(255)*	*Request priority*	*Priorità*	*Urgenza*
pr_Programmability	*VARCHAR(255)*	*Request programmability*	*Programmabilità*	-
pr_ActivityType	*TEXT*	*Activity type to be standardized according to ISO*	*Attività*	*Categoria O.I. per Ticket*
pr_ActivityDiscipline	*TEXT*	*Activity reference to be standardized according to ISO (e.g., 'electric')*	*Competenza*	-
ts_Request	*DATETIME*	*Date and time the request was open*	*Apertura*	*Data richiesta*
ts_OpeningTicket	*DATETIME*	*Date and time the request was taken in charge*	*Data presa in carico*	*Data apertura*
ts_OnSiteTechnicalControl	*DATETIME*	*Date and time of inspection*	*Data e ora sopralluogo*	*Registrazione sopralluogo*
ts_Closing	*DATETIME*	*Date and time the request was closed*	*Data e ora chiusura*	*Data chiusura*

1.2.3 Database "pre-processing"

Then, through a two-step process consisting of text pre-processing and cleaning after the field alignment of the respective databases shown in Section 1.2.2, it was possible to consolidate all intervention requests into a single, unified database of about 40k requests. As a result, this consolidated database bridges "unstructured" and "structured" data for each request, allowing to establish relationships between the textual description by end-users (who can have different levels of knowledge on maintenance issues) and key properties or structured fields, such as intervention priority, activity type, location, and timelines.

In particular, to enhance the efficiency and accuracy of text analysis, a structured pre-processing framework was applied to the unified dataset of requests:

- **Conversion to lowercase**. This step ensures consistency in the dataset by making all text lowercase, eliminating case sensitivity in the analysis. *Library: pandas.*
- **Removal of symbols, numbers, special characters, non-alphabetical characters, and non-ASCII characters** (e.g., #, @, 0-9, &, $). These elements do not contribute to the semantic content of the text. Their elimination ensures a cleaner dataset for processing. *Libraries: pandas, re* (Regular Expression – Python *re* module).
- **Replacement of newline characters**: this step ensures that newline characters are removed, preventing potential issues in sentence continuity and ensuring the text is uniform for analysis. *Library: pandas.*

- **Elimination of words shorter than three letters**, which helps filter out acronyms, typing errors, and unnecessary linguistic noise. Short words often lack meaningful context and can distort statistical and machine learning analyses. *Library: re* (Regular Expression – Python *re* module).
- **Stopwords removal**, excluding common words that do not carry significant meaning in sentence structure, such as articles, prepositions, and conjunctions. This step helps focus on the most informative words in the text, improving the relevance of extracted features. *Library: NLTK* (Natural Language Toolkit).
- **Lemmatization**, reducing words to their base or dictionary form. This technique is crucial for consolidating different word variations into a single root form, ensuring that the same concept is not treated as multiple distinct entities. (e.g., *vedo*, *vedi*, *vediamo* → *vedere*). *Library: NLTK* (WorldNetLemmatizer).
- **Stemming**, reducing words to their root form by cutting off suffixes. Unlike lemmatization, stemming follows heuristic rules and may produce non-dictionary words, but it can be useful in certain computational tasks where reducing dimensionality is a priority. (e.g., *vedo*, *vedi*, *vediamo* → *ved*). *Library: NLTK* (SnowballStemmer).

This pre-processing workflow was implemented to standardize textual content, minimize redundancy, and optimize the dataset for further text mining tasks such as topic modeling, classification, or sentiment analysis. Moreover, these different techniques were combined in various ways to identify the most effective sequence. The analysis revealed that the optimal combination included all steps, except for lemmatization, which was found to be less beneficial for the tasks.

1.2.4 Methods for the analysis of current status of building health

The analysis of current status of building health has been performed implementing calculation methods of selected KPIs directly in Python. Specifically, the analysis was performed using **DataSpell 2023.3.4** (JetBrains) with **Python 3.11.7**. The data outputs were organized through histograms, with values presented as percentages to provide a clear comparison of the metrics across different categories. Aggregation of the data was performed at multiple levels, based on predefined spatial areas and activity types. For the spatial analysis, the aggregation was conducted across all levels defined by the *IsPointOf* attributes, *pr_ActivityType*, and *pr_Priority* to ensure that critical aspects of the building health status were highlighted based on the established criteria. The actions performed in the code were facilitated by the use of two primary Python libraries. **Pandas** library has been used for data manipulation (e.g., *value_counts()*, *apply()*, *concat()*, *groupby()*, etc.), which is essential for operations like aggregating data, and managing columns and rows. **Matplotlib** has been used for creating bar charts.

In addition, sentiment and emotion analysis are performed using **VADER (Valence Aware Dictionary and sEntiment Reasoner)** and **FEEL_IT**, which are key libraries for analyzing textual data. **VADER** is a widely used tool for sentiment analysis, particularly effective for analyzing short texts like social media posts and reviews. It calculates sentiment scores based on the valence (positive or negative) of words in a sentence using four key metrics. The first metric, *positive*, represents the proportion of the text conveying a positive sentiment, while the *negative* metric indicates the part of the text with a negative sentiment. The *neutral* metric captures the proportion of the text that doesn't express any clear sentiment. Finally, the *compound* score is a composite metric that aggregates all the sentiments, ranging from -1 to +1. A score of -1 represents a highly negative sentiment, +1 indicates a highly positive sentiment, and a score of 0 suggests a neutral sentiment. For this study, the *vaderSentiment_ita* library was used, which is an adaptation of VADER tailored for Italian texts. Furthermore, **FEEL_IT** employs machine learning models to identify and classify emotions (i.e., *joy*, *sadness*, *anger*, and *fear*) in the text. This library provides a more refined understanding of the emotional tone behind the text, complementing the sentiment analysis performed by VADER.

1.2.5 Methods for predictive methods based on ML

In the same "how-to" perspective, Machine Learning (ML) [9] have been exploited by developing a reliable prediction model for automatic assignment of intervention features, starting from textual (thus "unstructured") description of maintenance requests by end-users. **Errore. L'origine riferimento non è stata trovata.** offers an outline of the whole process, comprising its correlation with database alignment and pre-processing. In particular, the automatic prediction of requests priority has been selected in view of the utility for decision makers, but the same process reported above can be applied for other features to be predicted.

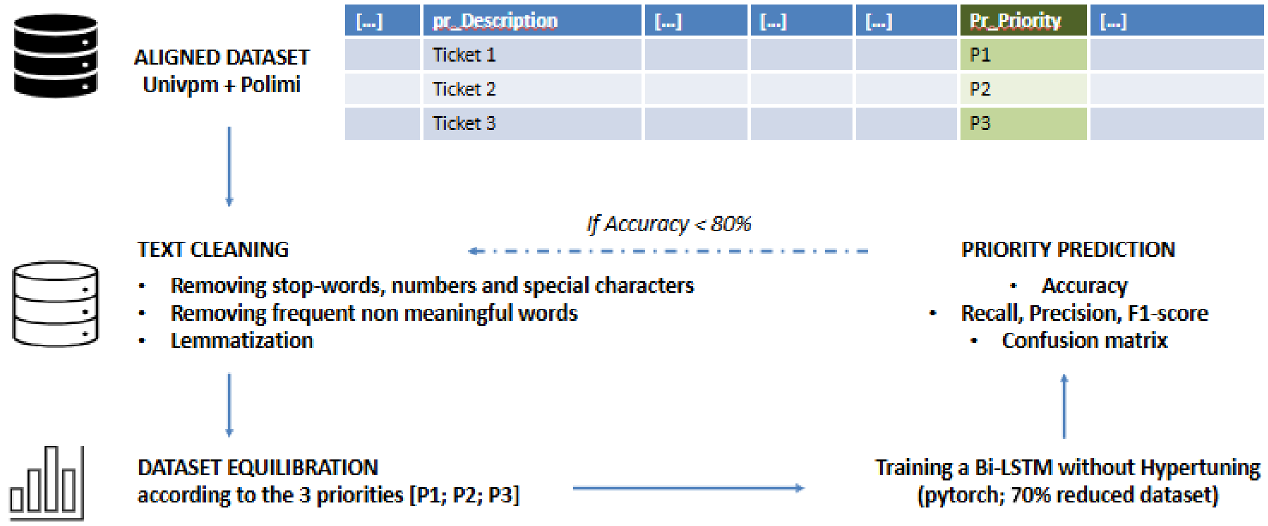
Once the database has been "cleaned" according to Section 1.2.3 operations for text pre-processing, the dataset has been preliminary equilibrated considering the identified classes for each feature to be predicted (P1, P2, and P3 bin, ascending order of urgency). The equilibration process has been performed to calibrate the dataset before starting with the training process and to reduce biases related to the composition of the input database in case of unbalanced classification conditions.

Then, a recurrent neural network architecture was selected, focusing specifically on a Bidirectional Long Short-Term Memory (Bi-LSTM) model. Bi-LSTM was chosen for its ability to process textual descriptions, leveraging context from forward and backward directions [21]. In the context of maintenance requests, this is particularly advantageous, as the semantic and temporal relationships within the textual descriptions can significantly influence the prediction accuracy. The Bi-LSTM model was trained using pre-processed data from the aligned database by combining consolidated pre-processing methods (focused on end-user textual requests) to ensure high-quality input [2], performing hyperparameter optimization, and testing different languages and tools to leverage computational efficiency (Python: Tensorflow, Pytorch). In particular, the models' effectiveness has been evaluated through the following indicators to outline the reliability of the prediction:

- **Accuracy**: the number of elements correctly classified concerning the total number of elements;
- **Recall**: the ratio of the number of elements correctly classified to the number of known elements in each class;
- **Precision**: the ratio of the number of elements correctly classified to the total predicted in each class;
- **F1-score**: the harmonic mean between both precision and recall;
- **Confusion matrix:** to assess misclassifications.

Various configurations were tested by modifying the cleaning procedure, the use of data equilibration, and different training-validation-testing splits (80:20, 70:30, and 70:20:10), with implementations carried out in Python. The **Python-based script** using Tensorflow and keras libraries has been preferred due to its computational speed, accuracy results, and model compatibility (*onnx* and *h5* formats). The final **Bi-LSTM RNN** model was tested with and without an overfitting regularizer layer, typology of loss function (focal or categorical cross-entropy), and boosting a hyper tuning process of the following parameters: embedding size, hidden units, dense units, dropout rate, and optimizer. The combination with the highest accuracy has been then selected as the final one to be hence implemented in the maintenance microservice for WP5 application (within the online tool). To check the model reliability for testing, the original database undergoes the whole pre-processing and ML process using the best combination but dividing again it (without database equilibration) into training, validation and test according to 70:20:10.

Figure 1. General process for prediction of requests' features, applied within the specific context of priority prediction.



2 Results

2.1 Key Performance Indicators

2.1.1 Complete KPIs list

Table 2 resumes all the KPIs proposed by DigitMan, according to general SMART concepts. In general terms, the unified database supports the calculation of three main categories of Key Performance Indicators (KPIs), designed to provide detailed insights into maintenance requests and their management.

Table 2: Overview of the proposed KPIs on maintenance requests. "Filters" columns are applied to Number of Requests, Percentage, and Percentile indicators. Each KPI can be also evaluated after normalization with respect to the location surfaces (i.e., room, floor, building).

		Output			Filter		
KPI (ID)	**KPI (name)**	**Number of Requests [n]**	**Pctg [%]**	**Percentile analysis (5, 50, 95)**	**by Location (*isPointOf*)**	**by Activity**	**by Priority**
M-1	*Requests*	*x*	*x*		*x*	*x*	*x*
M-2.1	*TimeToReply [OpeningTicket-Request]*			*x*	*x*	*x*	*x*
M-2.2	*TimeToControl [OnSiteTechnical Control – OpeningTicket]*			*x*	*x*	*x*	*x*
M-2.3	*TimeToClose [Closing – Request]*			*x*	*x*	*x*	*x*
M-2.4	*TimeToActivate [KPI2.2 – KPI2.1]*			*x*	*x*	*x*	*x*
M-2.5	*TimeToExecute [KPI2.3 – KPI2.2]*			*x*	*x*	*x*	*x*
M-3	*Priority*		*x*		*x*	*x*	

		Output			Filter		
KPI (ID)	KPI (name)	Number of Requests [n]	Pctg [%]	Percentile analysis (5, 50, 95)	by Location (*isPointOf*)	by Activity	by Priority
M-4	*Programmability*		*x*		*x*	*x*	
M-5.1	*VaderScore_users*			*x*	*x*	*x*	*x*
M-5.2	*VaderScore_tech*		*x*		*x*	*x*	*x*
M-6.1	*Emotion_users*			*x*	*x*	*x*	*x*
M-6.2	*Emotion_tech*		*x*		*x*	*x*	*x*

2.1.2 Short KPIs list

On this basis, KPIs have been refined to derive a limited number which can provide an overview of the whole building health status by request typology and position within the building stock.

KPI M-1, which details the number of maintenance requests collected, presented in various forms (absolute numbers [-], percentages [%], percentiles [%], and number of requests normalized by subject area [1/m^2]) and filtered based on request Activity type, classified according to the updated OmniClass Table 22, and intervention Location, which have been provided at the macro (each building or building complex in the whole building stock), meso (e.g. floor), and micro (i.e., specific room) scales.

KPI M-2, as time performance metrics measuring the time implied to (M-2.1) execute (that is the duration between the on-site inspection and the completion of the intervention) and (M-2.2) complete (that is the total time from the ticket opening and the completion of the intervention) organized by Activity type.

KPI M-3, to evaluate the priority level of each request, categorized into three urgency levels: P1 (High urgency), P2 (Medium urgency), and P3 (Low urgency).

Preliminary results shown in *Figure 2*) point out that for both UNIVPM and POLIMI, over 50% of the reported maintenance requests (KPI M-1) concern issues related to building systems and widespread facilities, specifically electrical, HVAC, and plumbing systems (respectively, codes 04, 11, and 14 in the DIGITMAN Activity Types list derived by OmniClass Table 22). Consequently, the time performance metrics (see Figure 3) are predominantly influenced by the handling of these types of interventions, also in view the complexity of these systems in technological and intervention perspectives. In this sense, KPI M-2.1 and KPI M-2.2 median values reveals slight differences between the two pilot cases for request types, which could be also due to organizational issues within maintenance staff. For example, electrical issues are resolved in just under one day at UNIVPM, com-pared to just over one day at POLIMI. Similarly, when analyzing KPI M-3 (see Figure 4), the aforementioned three Activity types account for more than 50% of high-urgency interventions. Notably, at POLIMI, HVAC-related issues represent a peak, constituting over 80% of interventions classified as P1.

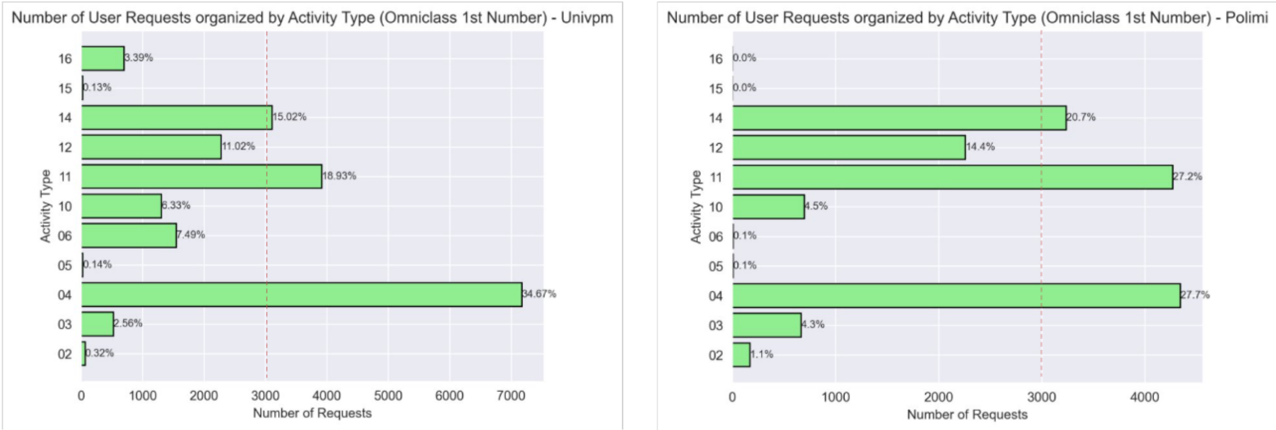


Figure 2. Statistics on Maintenance KPI M-1 for UNIVPM and POLIMI pilots.

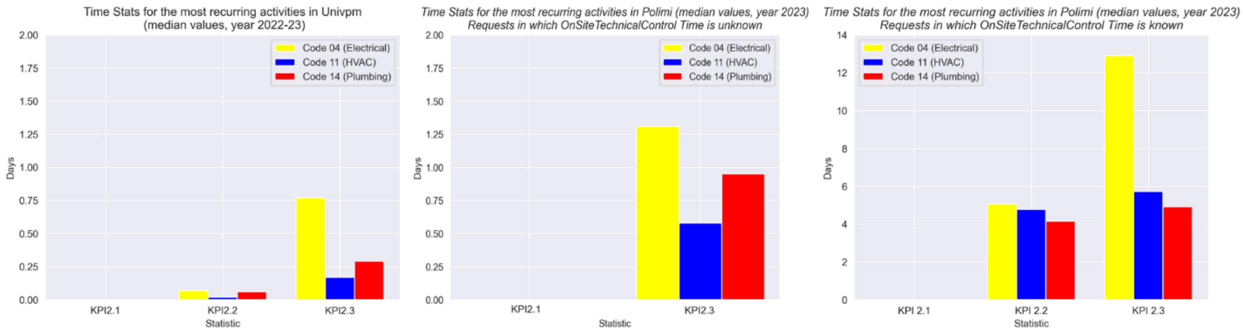


Figure 3. Statistics on Maintenance KPI M-2 for UNIVPM and POLIMI pilots. Filters: median values, years 2022-23. Data for POLIMI are distinguished whether the OnSiteTechnicalControl is indicated or not.

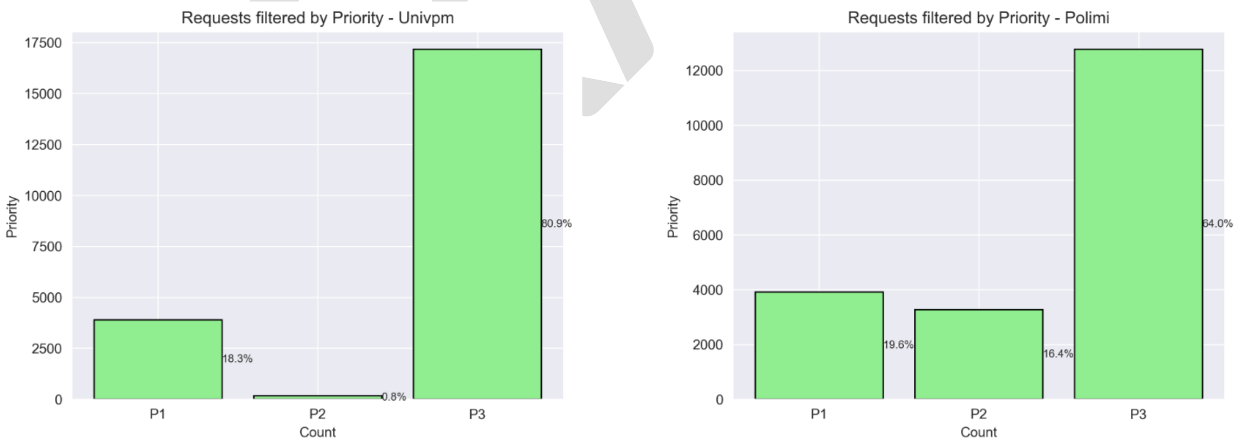


Figure 4. Statistics on Maintenance KPI M-3 for UNIVPM and POLIMI pilots (P1: High urgency, P2: Medium urgency, P3: Low urgency).

2.1.3 KPIs arrangement for WP5 multicriterial purposes

A predictive model has been developed to evaluate the relevance of maintenance requests across different areas (e.g., *rl_IsPointOf_Building* within a broader *rl_IsPointOf_BuildingStock*) based on the combination of relevant Key Performance Indicators derived in section 2.1.1. These KPIs capture various dimensions of maintenance demand and efficiency, allowing for comparative assessment across different areas, and include:

- **Activity Percentage filtered by Location** (KPI1 M-1 in Table 2): This represents the percentage of maintenance requests originating from a specific area with respect to the total number of requests. It quantifies the impact of a given building on overall maintenance demand, with values ranging from 0 to 100.
- **Priority Percentage** (KPI1 M-3 in Table 2): This metric indicates the percentage of requests classified under the highest priority category (P1), with values ranging from 0 to 100.
- **Programmability Percentage** (KPI M-4 in Table 2, applicable only to UNIVPM): this indicator measures the percentage of requests classified under the highest programmability category (P1), with values ranging from 0 to 100.
- **Time to Close** (KPI M-2.3 in Table 2): This indicator represents the cumulative duration of maintenance interventions in a specific area, expressed as a percentile within the dataset, thus ranging from 0 to 100. It provides an indication of the overall maintenance burden.
- **Time to Execute** (KPI M-2.5 in Table 2): This indicator measures the execution time of maintenance interventions in a specific area, expressed as a percentile within the dataset, thus ranging from 0 to 100. It reflects operational efficiency and responsiveness in maintenance execution.

The KPIs can be structured either by spatial areas or by Omniclass activity categories, enabling a detailed assessment of maintenance requests across different intervention types. Further weighting adjustments can be applied to balance their relative importance while potentially excluding categories with negligible impact. Expert judgment will be used to prioritize specific maintenance tasks.

2.2 Predictive methods: results

Preliminary results demonstrate that, among the tested pre-processing methods, the combination of stopwords removal, special character elimination, and stemming proved to be the most effective pre-processing procedure, even though with marginal gains compared to alternative methods (e.g. lemmatization). The best pre-processing framework also includes the following operations: no dataset equilibration (which is incorporating all 40k requests); removal of proper names and frequent meaningful words unrelated to priority prediction (such as greetings, acknowledgments, and titles); a 70-20-10 split for training, validation, and testing; absence of an overfitting regularization layer; and the adoption of focal loss. Nevertheless, it is worth noting that, considering the given database, alternating or combining different pre-processing methods seem to have a relatively low impact on the accuracy of the priority prediction. Then, the optimal configuration obtained from the hyper tuning considers the following parameters: embedding size = 150, hidden units = 64, dense units = 64, dropout rate = 0.5, optimizer = adam. Final reliable results are obtained considering the evaluation parameters (accuracy = 0.87; for the three priority classes P1, P2 and P3: F1-score= 0.76; 0.62; 0.93, precision= 0.80; 0.71; 0.90, recall= 0.72; 0.55; 0.95, and the confusion matrix shown in Figure 5).

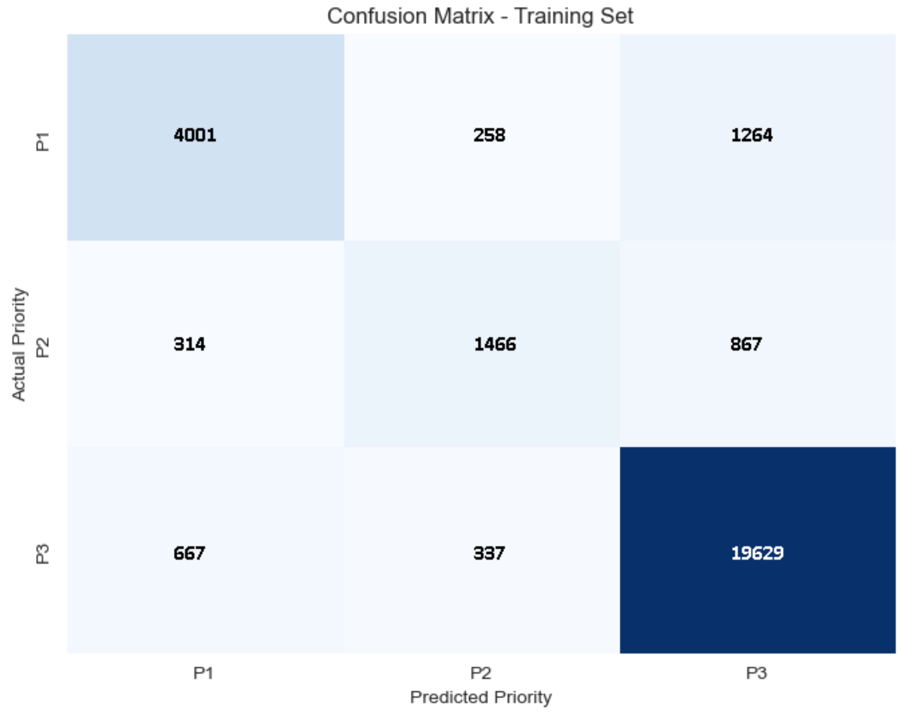


Figure 5. Confusion matrix – Training set.

3 Final remarks

The definition of KPIs on the maintenance issues and the development of the related analysis an prediction tools contribute to the analysis of the building stock for "how-to" decision making tasks. In view of the KPI structure, decision makers can:

- Simply and quickly check the current health status of the building, also tracing them over time, space and typology. This ensures detecting criticalities of the maintenance needs;
- Obtaining direct support for the quick assignment of request features, thanks to ML approaches and the automatic prediction of intervention typology, priority and programmability. In this research step, efforts have been focused on priority prediction, but the same methods and tools could be easily applied both to typology (OMNICLASS-based) and programmability.

In particular, concerning the building health status, by organizing and integrating these attributes, the database facilitates advanced analytics to mainly support "how-to" logics, such as recognizing patterns in user requests, understanding the impact of building occupancy profiles, and identifying probability distributions for maintenance work orders based on their type, urgency, and other factors. The unified structure not only improves data traceability but also enables the application of predictive maintenance methodologies, supporting the overarching goal of optimizing resource allocation and response efficiency within the pilot cases.

The automatic assignment of request priorities can then support facility management contractors in timely and accurate identification of necessary actions, reducing the efforts of technicians in repetitive and time-consuming actions. Moreover, ML approaches could be also used in further research steps to match classification of requests under different building conditions, such as intended use and occupancy rate over time, thus forecasting efforts for maintenance tasks under alternative scenario conditions, in a "what-if" logic.

The future implementation of KPIs into assessment metrics could mainly rely on reduced KPIs list defined in Section 2.1.2 as numerical leading KPIs within the same range. Basic assumptions for combination could be based on possible association of weights to health status KPIs, according to expert judgment or Analytical Hierarchy process techniques, as long as they can provide rapid index and unique metrics to assess the building health status as a whole, balancing the impact of each assessed components of short list of KPIs (see Section 2.1.3) on maintenance performance. These tasks will be completed in WP5 to merge them with other "how-to" microservices. About features prediction, the trained Bi-LSTM will be implemented in the WP5 service to directly predict requests features, thus avoiding additional training and validation tasks ongoing. It could be assumed that the decision-makers would be assisted in the management of a given new request / a group of new requests since the service can suggest to them the predicted class for the given features, allowing final supervision to accept or not these outputs. This can ensure higher acceptability from decision makers, who will be aware of automated assignment tasks and could modify them depending on additional constraints which are not included in the model. Moreover, additional re-training procedures could be explored in case the basic database will be significantly enlarged.

Finally, the proposed approach selects a limited but reliable number of KPIs which can be associated with spatial elements within the building model, e.g. using BIM-based or topological approaches. This imply that both health status and prediction methods would be linked to specific elements in the whole graphs of the building (spaces, components, activities) allowing decision makers to filter and locate them by building stock element. This perspective would be implemented in WP5 microservices to link all the information from microservices within the whole analysis and decision making dashboard for facility managers.

4 References

1. Zhao J, Feng H, Chen Q, Garcia de Soto B (2022) Developing a conceptual framework for the application of digital twin technologies to revamp building operation and maintenance processes. Journal of Building Engineering 49:104028. https://doi.org/10.1016/j.jobe.2022.104028

2. Gunay HB, Shen W, Yang C (2019) Text-mining building maintenance work orders for component fault frequency. Building Research & Information 47:518–533. https://doi.org/10.1080/09613218.2018.1459004

3. Burak Gunay H, Shen W, Newsham G (2019) Data analytics to improve building performance: A critical review. Automation in Construction 97:96–109. https://doi.org/10.1016/J.AUTCON.2018.10.020

4. Silva A, de Brito J (2019) Do we need a buildings' inspection, diagnosis and service life prediction software? Journal of Building Engineering 22:335–348. https://doi.org/10.1016/j.jobe.2018.12.019

5. D'Orazio M, Di Giuseppe E, Bernardini G (2022) Automatic detection of maintenance requests: Comparison of Human Manual Annotation and Sentiment Analysis techniques. Automation in Construction 134:104068. https://doi.org/10.1016/j.autcon.2021.104068

6. Abuimara T, Hobson BW, Gunay B, O'Brien W, Kane M (2021) Current state and future challenges in building management: Practitioner interviews and a literature review. Journal of Building Engineering 41:102803. https://doi.org/10.1016/j.jobe.2021.102803

7. Banfi F, Brumana R, Salvalai G, Previtali M (2022) Digital Twin and Cloud BIM-XR Platform Development: From Scan-to-BIM-to-DT Process to a 4D Multi-User Live App to Improve Building Comfort, Efficiency and Costs. Energies 15:. https://doi.org/10.3390/en15124497

8. Wang W, Guo H, Li X, Tang S, Xia J, Lv Z (2022) Deep learning for assessment of environmental satisfaction using BIM big data in energy efficient building digital twins. Sustainable Energy Technologies and Assessments 50:101897. https://doi.org/10.1016/j.seta.2021.101897

9. McArthur JJ, Shahbazi N, Fok R, Raghubar C, Bortoluzzi B, An A (2018) Machine learning and BIM visualization for maintenance issue classification and enhanced data collection. Advanced Engineering Informatics 38:101–112. https://doi.org/10.1016/j.aei.2018.06.007

10. Bortolini R, Forcada N (2020) Analysis of building maintenance requests using a text mining approach: building services evaluation. Building Research & Information 48:207–217. https://doi.org/10.1080/09613218.2019.1609291

11. Dong B, Yan D, Li Z, Jin Y, Feng X, Fontenot H (2018) Modeling occupancy and behavior for better building design and operation—A critical review. Building Simulation 11:899–921. https://doi.org/10.1007/s12273-018-0452-x

12. Yakhou N, Thompson P, Siddiqui A, Abualdenien J, Ronchi E (2023) The integration of building information modelling and fire evacuation models. Journal of Building Engineering 63:105557. https://doi.org/10.1016/j.jobe.2022.105557

13. Tang S, Shelden DR, Eastman CM, Pishdad-Bozorgi P, Gao X (2019) A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends. Automation in Construction 101:127–139. https://doi.org/10.1016/j.autcon.2019.01.020

14. Mirahadi F, McCabe B, Shahi A (2019) IFC-centric performance-based evaluation of building evacuations using fire dynamics simulation and agent-based modeling. Automation in Construction 101:1–16. https://doi.org/10.1016/j.autcon.2019.01.007

15. Siddiqui AA, Ewer JA, Lawrence PJ, Galea ER, Frost IR (2021) Building Information Modelling for performance-based Fire Safety Engineering analysis – A strategy for data sharing. Journal of Building Engineering 42:102794. https://doi.org/10.1016/j.jobe.2021.102794

16. Zhang J, Jiang S, Qi X (2024) Enhancing fire safety with Improved Risk Index and BIM in building evacuation. Engineering, Construction and Architectural Management. https://doi.org/10.1108/ECAM-09-2023-0923

17. Yenumula K, Kolmer C, Pan J, Su X (2015) BIM-Controlled Signage System for Building Evacuation. Procedia Engineering 118:284–289. https://doi.org/10.1016/j.proeng.2015.08.428

18. Galea ER, Xie H, Deere S, Cooney D, Filippidis L (2017) Evaluating the effectiveness of an improved active dynamic signage system using full scale evacuation trials. Fire Safety Journal. https://doi.org/10.1016/j.firesaf.2017.03.022

19. Malagnino A, Corallo A, Lazoi M, Zavarise G (2022) The Digital Transformation in Fire Safety Engineering over the Past Decade Through Building Information Modelling: A Review. Fire Technology 58:3317–3351. https://doi.org/10.1007/s10694-022-01313-3

20. Doran GT (1981) There's a S.M.A.R.T. way to write managements's goals and objectives. Management Review., 70(11). 35–36

21. Jang B, Kim M, Harerimana G, Kang S, Kim JW (2020) Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism. Applied Sciences 10:5841. https://doi.org/10.3390/app10175841