

MACHINE LEARNING-BASED DECISION SUPPORT FRAMEWORK FOR BIM COMPONENT SPECIFICATION

SUBMITTED: December 2025

PUBLISHED: May 2026

EDITOR: Bimal Kumar

DOI: [10.36680/j.itcon.2026.025](https://doi.org/10.36680/j.itcon.2026.025)

İpek Geç, Architect

Architectural Design Computing Program, Graduate School, Istanbul Technical University, Istanbul, Türkiye & ENKA Construction, ENKA Design Center, Istanbul, Türkiye

<https://orcid.org/0009-0004-2683-0886>

gec@itu.edu.tr

Orkan Zeynel Güzelci, PhD

Department of Interior Architecture, Faculty of Architecture, Istanbul Technical University, Istanbul, Türkiye & Faculty of Architecture + DFL/CEAU, University of Porto, Porto, Portugal

<https://orcid.org/0000-0002-5771-4069>

guzelci@itu.edu.tr & oguzelci@arq.up.pt

SUMMARY: Building Information Modeling (BIM) environments contain structured, data-rich models, yet the design logic embedded within them is rarely reused beyond individual projects. Although BIM systems capture relationships between spatial configurations and building components, this knowledge typically remains project-specific and is not systematically utilized in future design processes. This study introduces a machine learning (ML)-based computational framework that predicts door specification attributes from room-to-room spatial transitions, aiming to support data-informed decision-making in design. In this context, BIM is approached not only as a modeling environment but also as a source of transferable design knowledge. A dataset of 5,763 door instances was compiled from three Revit-based projects representing different building typologies: a hospital, an office, and a recreational facility. Spatial transitions were encoded using a Bag-of-Words (BoW) representation, and Random Forest algorithms were applied for classification and regression. Four scenarios were tested: (i) intra-project learning, (ii) cross-project learning, (iii) combined project learning, and (iv) domain-augmented retraining. To provide a benchmark, model performance was evaluated against simple baseline strategies, including majority-class prediction for classification and mean-value prediction for regression. Results showed strong performance for some targets within single projects, while others exhibited more moderate results. Performance decreased across projects due to inconsistent naming, class imbalance, and label mismatch, where the model encounters previously unseen categories. The findings demonstrate that archived BIM models can be transformed into predictive design intelligence, enhancing computational reasoning and efficiency within BIM-based workflows, while emphasizing the need for data standardization to achieve robust cross-project applicability.

KEYWORDS: Building Information Modeling, data-driven design, decision support systems, machine learning, spatial semantics.

REFERENCE: Geç, İ., & Güzelci, O. Z. (2026). Machine learning-based decision support framework for BIM component specification. *Journal of Information Technology in Construction (ITcon)*, 31, 561-583. <https://doi.org/10.36680/j.itcon.2026.025>

COPYRIGHT: © 2026 The author(s). This is an open access article distributed under the terms of the Creative Commons Attribution 4.0 International (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Building Information Modeling (BIM) has become a widely adopted methodology in the architecture, engineering, and construction (AEC) industry, enabling data-rich representations of buildings that extend across design, construction, and operation phases (Eastman et al., 2011). BIM has been widely recognized for its benefits in improving project performance, including cost reduction, time savings, and enhanced coordination across the project lifecycle (Bryde et al., 2013). While BIM systems offer advanced capabilities for managing geometric, semantic, and regulatory data (Succar, 2009; Becerik-Gerber & Kensek, 2010), BIM data has also been increasingly explored in relation to emerging digital technologies, including artificial intelligence, extended reality, digital twins, and robotics (Oesterreich & Teuteberg, 2016; Jin et al., 2026; Park et al., 2026).

However, despite these developments, BIM data is still often confined to individual project workflows, and the knowledge embedded in completed models is rarely structured or systematically reused to inform subsequent projects. Although BIM models capture object families, component properties, and spatial configurations as structured datasets, often organized through standardized schemas such as Industry Foundation Classes (IFC) (International Organization for Standardization, 2025), they are typically archived at the end of a project, leaving this embedded project logic largely unused.

Recent studies have also explored how BIM data can be accessed and interpreted through advanced query and language-based systems, enabling more flexible information retrieval and interaction with model content (Guo et al., 2025). Previous studies have also highlighted that, despite the increasing adoption of BIM, the reuse of information across projects remains limited. BIM data is often structured around project-specific standards, naming conventions, and workflows, which restrict its transferability and reuse in new design contexts (Volk et al., 2014; Pauwels et al., 2017). More recent research further emphasizes that, although BIM is increasingly integrated with emerging technologies such as artificial intelligence and digital twins, its role as a reusable knowledge base across projects is still underdeveloped (Boje et al., 2020). As a result, valuable design knowledge embedded in past projects is not systematically utilized, and similar design decisions are repeatedly made from scratch.

Taken together, these findings highlight a persistent limitation in current BIM practices. This limitation creates a critical gap between isolated project intelligence and transferable design knowledge that can inform future decisions. This study addresses that gap by proposing a machine learning (ML)-based approach for decision support that learns from architectural data embedded in completed BIM models and applies this learning to predict component specifications in future projects. In this sense, BIM can be understood not only as a modeling platform but as part of a broader shift toward digital buildings, where data-driven processes and integrated information systems play a central role in design and construction (Watson, 2011). Accordingly, BIM is reframed here as a dynamic knowledge system capable of evolving through accumulated experience, echoing Ackoff's (1989) notion of the transition from data to wisdom.

Two research questions guide this study:

- Can information embedded in past BIM projects be systematically transformed into a decision support mechanism based on predictive modeling for estimating component-level specifications in new projects?
- Can spatial-functional adjacency between rooms serve as a reliable basis for predicting component family configurations in new design contexts?

Within this context, the study explores how BIM-derived metadata can support decision-making through predictive modeling, while establishing a methodological foundation for future, more integrated implementations. To explore these questions, the study focuses on a specific and manageable domain: door families modeled in three different building typologies using Revit-based BIM workflows developed by a professional design office. Door families were selected as the study domain since they combine semantic richness with parametric diversity and exhibit strong dependence on spatial-functional adjacency (e.g., transitions, fire zones). These three projects, a hospital (Project 1), an office building (Project 2), and a recreational facility (Project 3), collectively provide a representative range of spatial configurations and design logic. Together, these models contributed 5,763 door family instances, offering a rich and diverse dataset for predictive analysis. Autodesk Revit is chosen as the BIM

platform due to its standardized data structure and widespread professional adoption (Krygiel & Nies, 2008; Kensek, 2014).

The proposed methodology follows a structured ML approach composed of three stages: (1) defining the problem scope and target component, (2) collecting and preprocessing door family data from Revit-based BIM models and encoding spatial features using a Bag-of-Words (BoW) representation derived from simple text processing techniques, (3) selecting a suitable learning algorithm and training the model under grouped cross-validation to ensure generalizability across room-to-room transitions.

The approach is implemented as a standalone analytical pipeline operating on BIM-derived metadata from completed projects. It is not embedded within BIM authoring environments such as Autodesk Revit, but demonstrates how such data can be utilized for predictive modeling. The workflow uses previously completed BIM models as input and produces predictive outputs that can inform subsequent design decisions, while also providing a foundation for future integration into BIM-based design tools.

The main contribution of this study is to show that information embedded in archived BIM data, particularly the relationships between spatial configurations and component selections, can be used to estimate component-level properties through predictive modeling. While the current scope is limited to door families modeled in Autodesk Revit, the approach can be extended to other building components and different BIM environments.

2. BACKGROUND

Building Information Modeling (BIM) provides an integrated digital environment for modeling not only the geometry but also the functional, material, regulatory, and semantic properties of architectural components throughout a building's lifecycle (Eastman et al., 2011). Prior studies confirm that BIM usage is often project-specific, limiting opportunities for knowledge transfer across contexts (Azhar, 2011).

To overcome these limitations, researchers have investigated the integration of Decision Support Systems (DSS) into BIM workflows. DSS are interactive frameworks that support designers and stakeholders by processing structured data and generating context-aware suggestions (Power, 2002). Within the architecture, engineering, and construction (AEC) industry, DSS have been used to improve design efficiency and support more informed decision-making processes (Zhang & Issa, 2013; Marcher et al., 2020). In practice, most BIM-based DSS have relied on rule-based reasoning, optimization methods, or simulation-driven evaluations. Such systems have been particularly applied to domains such as energy performance assessment, cost estimation, and automated compliance checking (Solihin & Eastman, 2015). Ontology-driven DSS have also been explored to enable interoperability and consistency between projects (Zhang & Issa, 2013). However, both rule-based and ontology-driven systems are constrained by predefined knowledge bases and therefore have limited adaptability when exposed to diverse project contexts.

Advances in ML have created new opportunities to reinterpret BIM data as a dynamic training resource rather than a static archive. ML algorithms, particularly supervised learning techniques, can identify patterns in high-dimensional datasets and generate predictive models based on previous outcomes (Hastie et al., 2009). Unlike rule-based DSS, ML approaches are data-driven and adaptive, offering the potential to capture implicit design intelligence from archived projects.

Although the construction industry generates large volumes of digital data, the adoption of ML remains relatively limited; however, its use has been steadily increasing in recent years (Bilal et al., 2016). Within BIM research, machine learning has been explored across a range of applications, including predictive modeling approaches such as energy performance prediction (Magnavaca de Paula et al., 2026), as well as broader applications in construction monitoring, safety analysis, and operational management (Khallaf & Khallaf, 2021). More recent work includes deep learning for component classification, graph-based models that integrate contextual information, and structured representations (Pauwels et al., 2017), and ontology-aided semantic parsers for BIM model querying (Yin et al., 2023). These contributions demonstrate the potential of BIM as a data source for learning-based systems.

However, several gaps remain:

- Most ML applications in BIM focus on numerical or performance-related parameters, while semantic or design-oriented decisions at the component family level are less explored.

- The reuse of design knowledge embedded in past projects has not been systematically addressed, limiting knowledge transfer across contexts.
- Conceptual frameworks such as parametric design cognition (Oxman, 2017) and earlier work on design knowledge transfer (Gero, 1990), followed by more recent studies on semantic reasoning and knowledge representation (e.g., Pauwels et al., 2017), emphasize the importance of structured and interpretable design knowledge; however, BIM-based machine learning studies rarely operationalize these perspectives.

To address these gaps, this study focuses on door families as a representative test domain, examining whether room-to-room transitions can be utilized to capture and generalize design logic across different projects. By targeting architectural elements that are both semantically meaningful and parametrically structured, the study explores how ML can support decision-making within BIM environments.

3. METHODOLOGY

The methodology consists of structured stages, including (1) problem scoping, (2) data collection and feature encoding, (3) data distribution, (4) evaluation scenarios, (5) algorithm selection and model training, and (6) evaluation metrics, each designed to support predictive tasks based on architectural and spatial context.

3.1 Problem and target component

The problem is formulated as a supervised learning task at the level of individual door instances. Each instance is represented by input features derived from BIM data, consisting of spatial attributes (room-to-room transitions encoded as text). The prediction targets consist of both categorical variables, such as leaf material, leaf type, frame type, and fire/smoke rating, treated as classification tasks, and continuous variables, including door width and height, treated as regression tasks. Accordingly, the model learns the relationship between spatial-functional context and door specification attributes associated with door elements.

Door elements are used as the study domain due to their discrete and parameterized nature. In this context, the prediction targets correspond to door specification attributes at the type or instance level (e.g., material, type, and performance properties), rather than complete Revit-specific family definitions. These attributes can be interpreted more broadly as type-level or property-level information consistent with BIM standards such as IFC. In building design, elements such as walls and floors are typically defined through layered assemblies and continuous material systems. Doors, in contrast, are usually described through a set of discrete attributes, including leaf material, type, and fire resistance rating.

These attributes are not independent from construction standards or material considerations. However, in practice, they are often closely linked to the functional relationship between adjacent spaces. Requirements related to circulation, privacy, or fire separation frequently emerge at the interface between rooms and are directly reflected in door specifications.

3.2 Data collection and feature encoding

The dataset was gathered from three Revit-authored BIM projects developed by a professional design office, covering three distinct typologies: a hospital, an office, and a recreational facility. These projects were intentionally selected to represent diverse building types, allowing the model's predictive ability to be tested under varied spatial and functional conditions and to evaluate its potential for cross-project generalization.

The dataset includes 5,763 door instances in total, with 2,950 doors from the hospital project, 1,610 from the office project, and 1,203 from the recreational facility. The initial dataset sizes were 4,533 door instances for the hospital project and 1,457 for the recreational facility, while the office dataset was used without filtering. Entries with missing key fields (e.g., from/to room information or target attributes), duplicated records, or ambiguous labels were excluded during preprocessing. This resulted in the removal of approximately 35% of the hospital data and 17% of the recreational facility data.

Door family data were extracted from Revit schedules and exported into structured tabular form, with each row corresponding to a unique door instance enriched with both physical parameters and contextual metadata. Although

the dataset contains both spatial descriptors and door specification attributes (e.g., width, height, leaf material, leaf type, frame type, and fire rating), these variables were not all used as input features in the predictive models. In the implemented models, the only input feature was the room-to-room relationship encoded as text.

Similarly, categorical attributes such as leaf material, leaf type, frame type, and fire/smoke rating were selected exclusively as prediction targets in separate classification tasks. For each prediction task, a separate model was trained, and the corresponding target variable was explicitly excluded from the input feature set. The models relied solely on the encoded spatial transition feature (combined_room) as input. No target attribute was used to predict another target attribute, thereby preventing information leakage.

All door specification attributes were treated as prediction targets, with continuous variables (door width and door height) modeled via regression and categorical variables (leaf material, leaf type, frame type, and fire/smoke rating) modeled via classification, as shown in Table 1.

Table 1: Dataset headings and parameters.

LEVEL	ROOM NUMBER	ROOM NAME	FROM ROOM	TO ROOM	DOOR NUMBER	DOOR ROUGH WIDTH	DOOR ROUGH HEIGHT	LEAF MATERIAL	LEAF TYPE	FRAME TYPE	FIRE RATE & SMOKE REQUIREMENT
L01	107	OPERATING ROOM	OPERATING ROOM	CORRIDOR	D001	800	2000	ALUMINIUM	SLIDING	BLOCK FRAME	EI60
L03	134	OPERATING ROOM	OPERATING ROOM	CORRIDOR	D002	900	2000	WOODEN	ASYMMETRICAL	ADJUSTABLE FRAME	0
L01	177	STAFF ROOM	STAFF ROOM	CORRIDOR	D003	900	2100	WOODEN	SLIDING	FULL FRAME	EIS30
L03	171	CORRIDOR	CORRIDOR	WAITING AREA	D004	800	2100	GLASS	SLIDING	CORNER FRAME	EIS60
L01	137	TOILET	TOILET	ICU	D005	1000	2000	ALUMINIUM	SINGLE LEAF	CORNER FRAME	0
L03	123	STAFF ROOM	STAFF ROOM	TOILET	D006	800	2100	GLASS	ASYMMETRICAL	ADJUSTABLE FRAME	EI30
L01	171	CORRIDOR	CORRIDOR	STAFF ROOM	D007	1200	2100	STEEL	SINGLE LEAF	FULL FRAME	EI30
L01	197	CORRIDOR	CORRIDOR	EXAMINATION ROOM	D008	1200	2000	GLASS	DOUBLE LEAF	BLOCK FRAME	EIS60
L03	117	STAFF ROOM	STAFF ROOM	WAITING AREA	D009	1100	2200	WOODEN	ASYMMETRICAL	ADJUSTABLE FRAME	EI30
L05	128	STAFF ROOM	STAFF ROOM	NURSE STATION	D010	1000	2100	ALUMINIUM	ASYMMETRICAL	ADJUSTABLE FRAME	EI45
L01	127	ICU	ICU	TOILET	D011	1200	2000	WOODEN	SLIDING	CORNER FRAME	0

Spatial metadata was also incorporated through the from_room and to_room fields, along with a derived feature named combined_room, which represents room-to-room relationships between adjacent spaces (e.g., “patient room to corridor”). The “from_room” and “to_room” attributes originate from the BIM data structure and reflect how door instances are recorded in Revit schedules, rather than an intentional encoding of directionality. Although these fields imply a directional relationship, the Bag-of-Words (BoW) representation used in this study does not preserve word order. Therefore, the model does not explicitly learn directional transitions, but instead captures co-occurrence patterns between room types.

This representation aligns with the nature of many door specifications, which are often symmetric with respect to adjacent spaces (e.g., a door between a toilet and a corridor typically follows the same typology regardless of direction). While certain architectural conditions may introduce asymmetric requirements (such as access control or fire zoning), capturing such effects would require sequence-aware or graph-based representations, which are beyond the scope of this study.

To capture this spatial relationship in a machine-readable form, the combined_room feature was transformed using a Bag-of-Words (BoW) approach. BoW is a commonly used text representation method that encodes textual data based on the frequency of individual tokens, without considering their order or grammatical structure (Jurafsky & Martin, 2023). Similar approaches have been applied in BIM research to extract and classify semantic information from textual data using natural language processing techniques (Jung & Lee, 2019).

In this study, the combined_room field was vectorized using the CountVectorizer function. Each room-to-room relationship (e.g., “patient room to corridor”) was decomposed into its constituent tokens, and a sparse matrix representation was generated based on their occurrence across the dataset, as shown in Figure 1.

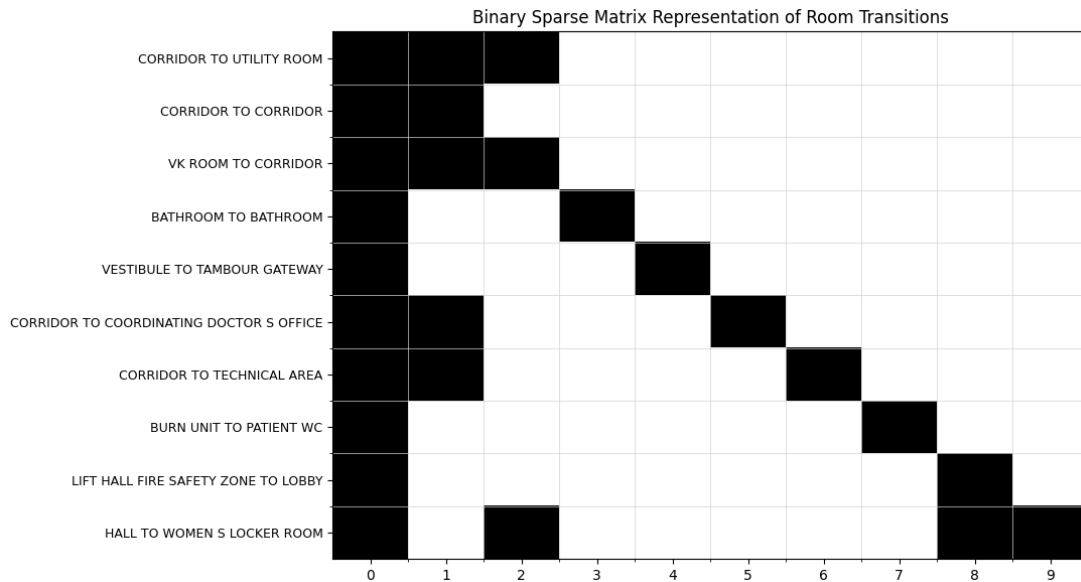


Figure 1: Sparse matrix representation of spatial transitions using Bag-of-Words encoding.

Each token is assigned a unique numerical index (0 → to, 1 → corridor, 2 → room, 3 → bathroom, 4 → vestibule, 5 → office, 6 → area, 7 → patient, 8 → hall, 9 → women), and input sequences are encoded accordingly. The table presents a subset of tokens from the overall vocabulary for illustration purposes. This representation demonstrates how spatial co-occurrence between rooms is transformed into machine-readable numerical features. The mapping is automatically learned from the dataset using a CountVectorizer implementation. For example, the sequence to, corridor, room is encoded as (0, 1, 2).

This transformation enables the model to capture recurring spatial patterns in room adjacencies and relate them to corresponding door design decisions. Rather than relying solely on explicit parameters, the model could implicitly learn architectural relationships such as circulation logic, functional zoning, and adjacency-driven constraints. In this sense, the feature representation extends beyond raw data encoding and begins to reflect how spatial relationships inform design decisions in practice.

Categorical fields were selectively cleaned and standardized depending on their role in the dataset. Specification-related attributes, such as material labels and fire/smoke ratings, were normalized by correcting formatting inconsistencies (e.g., “wood” vs. “wooden”, “EI60” vs. “EI-60”) and unifying equivalent representations. In contrast, room names were not standardized across projects, as they reflect project-specific naming conventions and functional distinctions. Entries with missing or ambiguous values were excluded during preprocessing. As a result, while inconsistencies within individual attributes were partially resolved, cross-project differences in naming and label definitions remained.

Each categorical target was defined as a finite label space derived from the cleaned dataset. For leaf material, labels correspond to distinct material categories (e.g., steel, aluminium, glass, wood). Leaf type includes door configuration types such as single, double, sliding, and asymmetrical. Frame type represents structural frame configurations (e.g., full frame, block frame, corner frame, adjustable frame). Fire and smoke rating labels follow standardized performance classes (e.g., EI30, EI60, EIS30), with “0” indicating no specific requirement.

To ensure consistency across projects, all labels were normalized through case standardization and manual harmonization of naming variations. Synonymous terms and inconsistent encodings were unified under a common label, and manufacturer-specific codes or project-specific abbreviations were excluded or mapped to their corresponding generic categories.

3.3 Data distribution

The distribution of categorical attributes was examined to provide a general understanding of the dataset structure prior to model training. Across the three projects, notable differences were observed in both class proportions and label availability.

At the project level, the hospital dataset is characterized by a more concentrated distribution, where a limited number of categories dominate most attributes. The office dataset presents a relatively similar but slightly more varied structure, while the recreational facility dataset exhibits a more heterogeneous composition, including a wider range of categories and less frequent labels.

Across all categorical targets (leaf material, leaf type, frame type, and fire rate & smoke requirement), class imbalance is present to varying degrees. In several cases, a small number of dominant categories account for the majority of instances, while other categories remain underrepresented. In addition, the label space is not fully consistent across projects, as some categories are present only in specific datasets. These variations indicate that the datasets are not fully aligned in terms of distribution or label definitions, which is an important consideration for cross-project learning scenarios. Figures 2–5 illustrate the distribution of categorical attributes across projects.

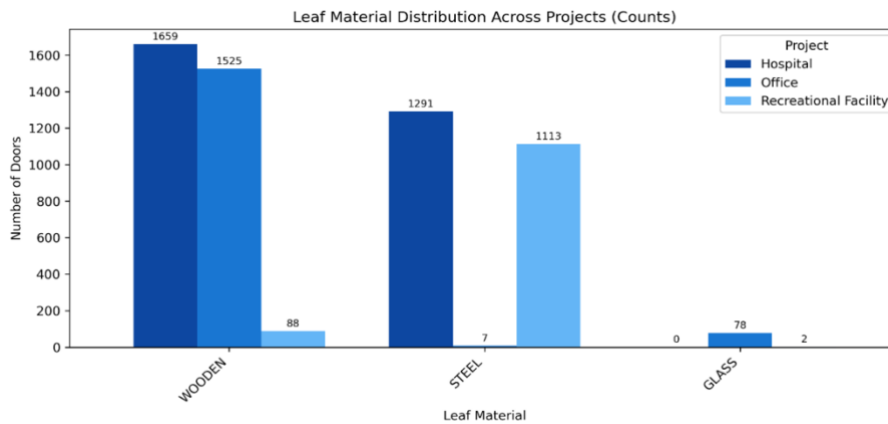


Figure 2: Distribution of leaf material categories across Project 1 (hospital), Project 2 (office), and Project 3 (recreational facility) (counts).

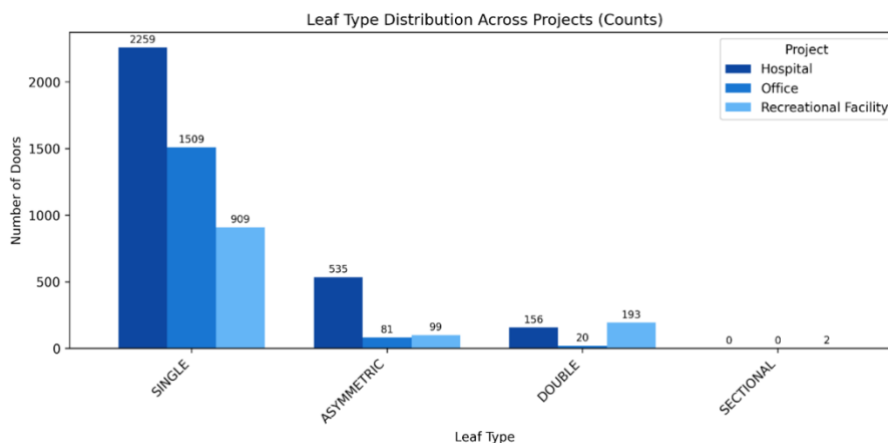


Figure 3: Distribution of leaf type categories across Project 1 (hospital), Project 2 (office), and Project 3 (recreational facility) (counts).

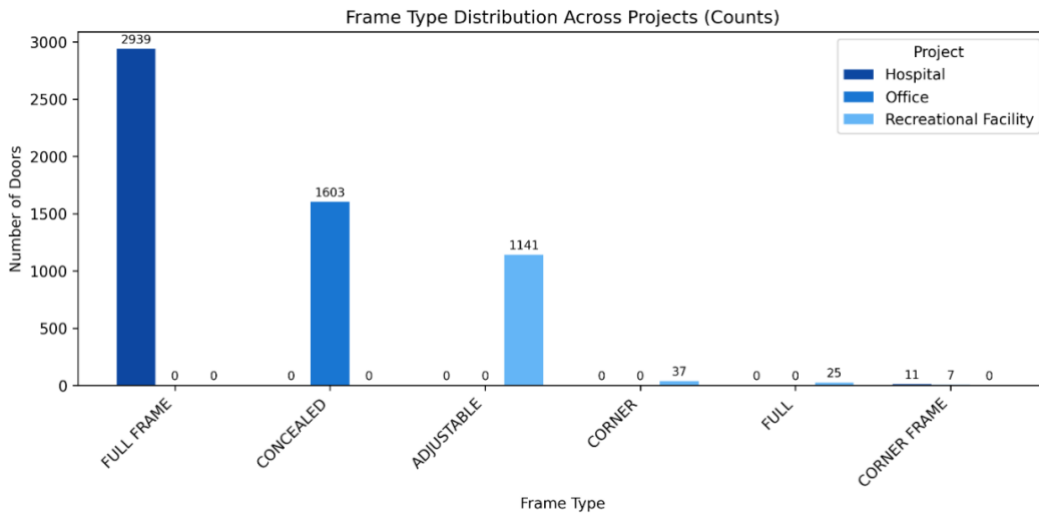


Figure 4: Distribution of frame type categories across Project 1 (hospital), Project 2 (office), and Project 3 (recreational facility) (counts).

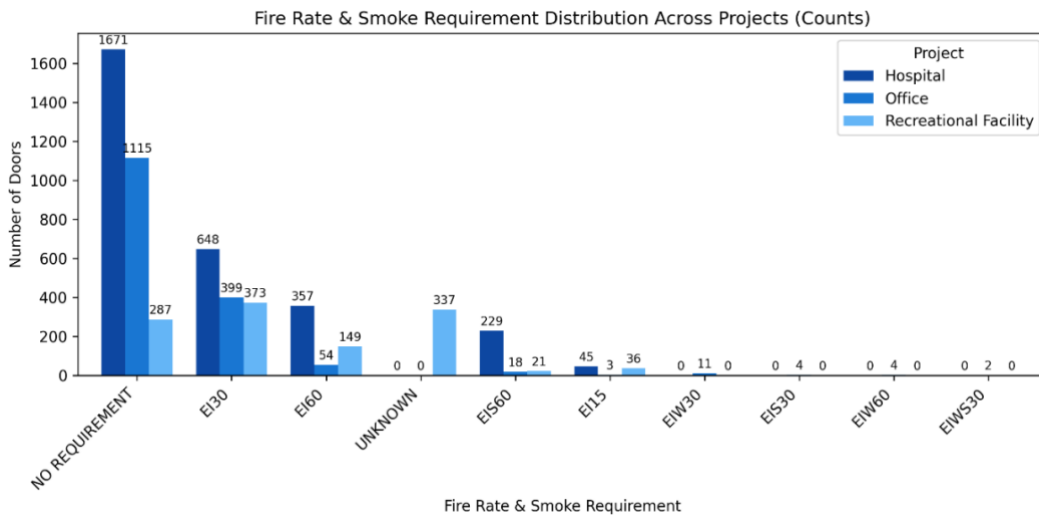


Figure 5: Distribution of fire rate and smoke requirement categories across Project 1 (hospital), Project 2 (office), and Project 3 (recreational facility) (counts).

3.4 Evaluation scenarios

To evaluate the model's ability to generalize across different spatial, typological, and semantic configurations, four experimental scenarios were defined:

- Scenario 1 - Intra-Project Learning (Project 1 (hospital) → Project 1 (hospital)): The model was trained and tested within the same architectural project to evaluate its predictive performance under consistent naming conventions, spatial patterns, and modeling standards.
- Scenario 2 - Cross-Project Generalization (Project 1 (hospital) → Project 2 (office)): The model was trained on Project 1 and tested on an entirely different building (Project 2), with no exposure to the target project data. This scenario evaluated the model's ability to transfer learned decision patterns across distinct typologies and contexts.
- Scenario 3 - Combined Project Generalization (Project 1 (hospital) + Project 2 (office) → Project 3 (recreational facility)): A combined training dataset from Projects 1 and 2 was used to develop a more

diverse learning base, which was then tested on a completely unseen project (Project 3). The goal was to assess whether exposure to multiple project types could enhance generalization.

- Scenario 4 - Domain-Augmented Retraining (Project 1 (hospital) + Project 2 (office) + partial Project 3 (recreational facility) → Project 3 (recreational facility)): To improve cross-project adaptability, the model was retrained using an augmented dataset that includes a subset of the target project, while the remaining data was reserved for testing. This setup tested whether limited exposure to the target context could significantly enhance predictive accuracy.

3.5 Algorithm selection and model training strategy

The study is structured around two prediction tasks: classification (leaf material, leaf type, frame type, fire/smoke rating) and regression (door width, door height). A single algorithm, Random Forest (Breiman, 2001), was employed across all targets. Random Forest builds ensembles of decision trees on bootstrap resamples and aggregates their predictions (majority voting for classification, averaging for regression). Rather than aiming to benchmark different algorithms, a single model family was used to ensure comparability across scenarios and to isolate the effect of data variability (e.g., distribution shifts, label inconsistencies, and project-specific conventions). This method provides a robust, low-tuning approach that can handle heterogeneous inputs, such as the bag-of-words representation of spatial context, while remaining interpretable across scenarios. Although linear models such as logistic regression or linear SVM are often effective for sparse Bag-of-Words features, their comparison falls outside the scope of this study and is considered for future work.

To avoid information leakage, models were grouped by the combined_room variable, ensuring that identical transitions never appear on both sides of a training–evaluation boundary. Scenario 1 (intra-project) used GroupKFold cross-validation (up to 10 folds depending on group cardinality) to estimate generalization within a single project. In contrast, Scenarios 2–4 used a single hold-out evaluation strategy, implemented through fixed project-based partitions. In Scenarios 2 and 3, training and testing sets were fully disjoint at the project level (cross-project evaluation). In Scenario 4, a subset of the target project was included in the training set for fine-tuning, while the remaining portion was reserved for testing. Therefore, only Scenarios 2 and 3 represent strictly disjoint cross-project settings, whereas Scenario 4 introduces a domain adaptation setup. Since the evaluation in Scenarios 2 and 3 was based on fully disjoint project partitions, cross-validation was not applicable in those cases. Instead, the primary challenge was domain shift and non-overlapping label sets. Textual features were represented with a bag-of-words (BoW) model applied to room transitions (combined_room). The encoded room transition features (combined_room), described in Section 3.2, were used as input to the Random Forest models.

Across all scenarios, six predictive models were defined: four Random Forest classifiers (leaf material, leaf type, frame type, fire/smoke rating) and two regressors (door width, door height). These six tasks were repeated for each of the four experimental scenarios, resulting in a total of 24 distinct models. In Scenario 1, each model was further trained across multiple cross-validation folds, producing additional fitted estimators internally, but still corresponding to the same six predictive tasks.

To provide a reference for model performance, simple baseline strategies were defined. For classification tasks, a majority-class baseline was used, where the most frequent class in the training data is always predicted. For regression tasks, a mean-value baseline was used, where the average target value is predicted for all instances. These baselines serve as a lower bound for evaluating whether the models learn patterns beyond dominant distributions.

All models were implemented in Python using the scikit-learn library. The experiments were conducted in a Google Colab environment, with data accessed from Google Drive. Standard implementations of Random Forest classifiers and regressors, as well as CountVectorizer, was used without modification. Default hyperparameters were used for all Random Forest models (e.g., n_estimators = 100, no maximum depth constraint), as provided by the scikit-learn library. Hyperparameter tuning was not the primary focus of this study; instead, default parameters were maintained to ensure a consistent baseline for evaluating the impact of spatial context across different scenarios. To ensure reproducibility, a fixed random seed (random_state = 42) was used across all model configurations.

3.6 Evaluation metrics

In each experimental scenario, categorical predictions are evaluated using row-normalized confusion matrices and standard classification metrics. Alongside the matrices, Accuracy, Precision, Recall, and F1-score are reported. Accuracy measures the proportion of correct predictions out of all predictions (1 = perfect, 0 = no correct predictions). Precision quantifies the correctness of positive predictions (higher is better), Recall measures coverage of actual positives (higher is better), and F1 is the harmonic mean of Precision and Recall (closer to 1.0 indicates stronger balanced performance). Given potential class imbalance, Macro-F1—the unweighted average of per-class F1—serves as the primary indicator of balanced performance, with Accuracy provided as a complementary summary.

Continuous targets, door width and door height are evaluated using Mean Absolute Error (MAE) and R^2 , and visualized via True-vs-Predicted scatter plots. MAE reports the average absolute deviation in millimeters, where lower values indicate better performance (0 = perfect prediction, higher = larger typical error). R^2 expresses explained variance, with 1.0 indicating a perfect fit, values near 0 corresponding to mean-level prediction, and negative values reflecting performance worse than predicting the mean. In the scatter plots, systematic under- or over-estimation appears as points consistently below or above the diagonal, while greater dispersion around the diagonal indicates larger random error. All regression metrics are computed on the scenario-specific test partitions defined by the split protocol.

4. RESULTS AND FINDINGS

This section presents the evaluation results of the predictive models under different testing scenarios, followed by an interpretation of their performance and limitations. The same learning algorithm (Random Forest) was consistently used across all experiments; however, the model was retrained separately for each scenario using the corresponding training and testing datasets. In this way, each case represents an independent evaluation of the approach under varying project contexts rather than the application of a single pre-trained model.

4.1 Results for Scenario 1: Intra-Project Learning (Project 1 (hospital) → Project 1 (hospital))

Within-project learning on the hospital dataset establishes a strong baseline because naming conventions, spatial patterns, and modeling standards are fully consistent across train and test splits. This makes Scenario 1 the “best-case” setting, reflecting the practical upper bound of the approach. Classification is highest for leaf material (Accuracy = 0.927; Macro-F1 = 0.926; Precision = 0.928; Recall = 0.927) and remains robust for leaf type (0.905; 0.789; 0.907; 0.905). Frame type yields near-perfect Accuracy (0.996) but a markedly lower Macro-F1 (0.576), indicating that majority classes dominate the metric and rare classes are seldom predicted correctly. Fire rate and smoke requirement is more difficult (0.817; 0.622; 0.804; 0.817), reflecting the larger and less evenly distributed label set (Figure 6; Figure 7). Regression performs well for door width (MAE = 93.5 mm; R^2 = 0.620) but trivially for door height (MAE = 0.0 mm; R^2 = 1.000) because the hospital project contained uniform door height values across all instances (2160 mm) (Figure 8). This apparent perfect performance does not reflect actual predictive capability but is instead a result of zero variance in the dataset, where the model effectively learns a constant value rather than a meaningful relationship. Therefore, this result should be interpreted as a data artifact rather than a valid indicator of model accuracy and is excluded from positive performance evaluation. Thus, Scenario 1 shows that the model can reliably reproduce within-project decision patterns, but its performance is constrained by class imbalance and limited label variation rather than algorithmic limits.

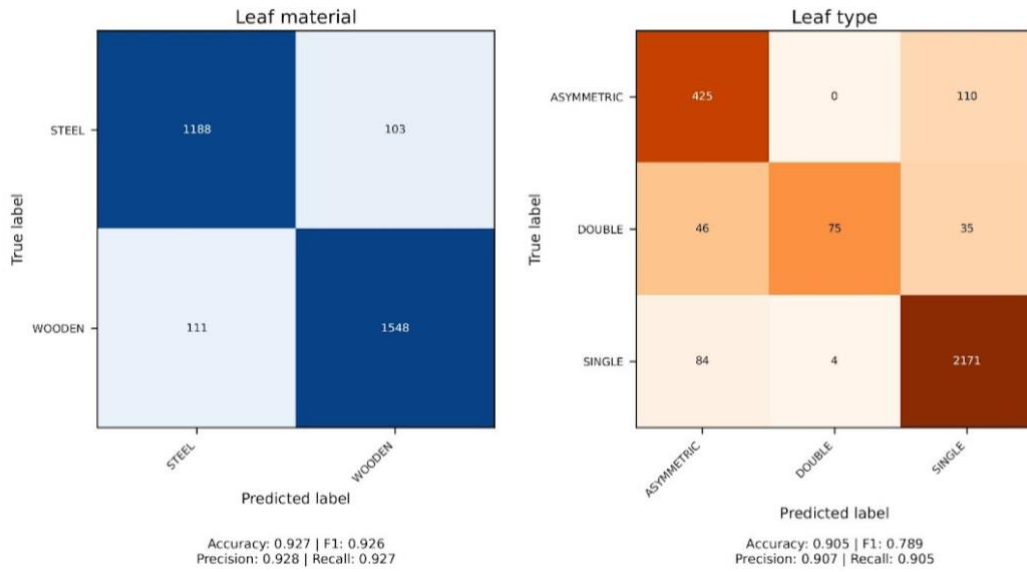


Figure 6: Scenario 1 – confusion matrix results. Cell colors represent row-normalized values (0–1), where darker shades indicate higher prediction accuracy within each class, while the numbers denote raw counts.

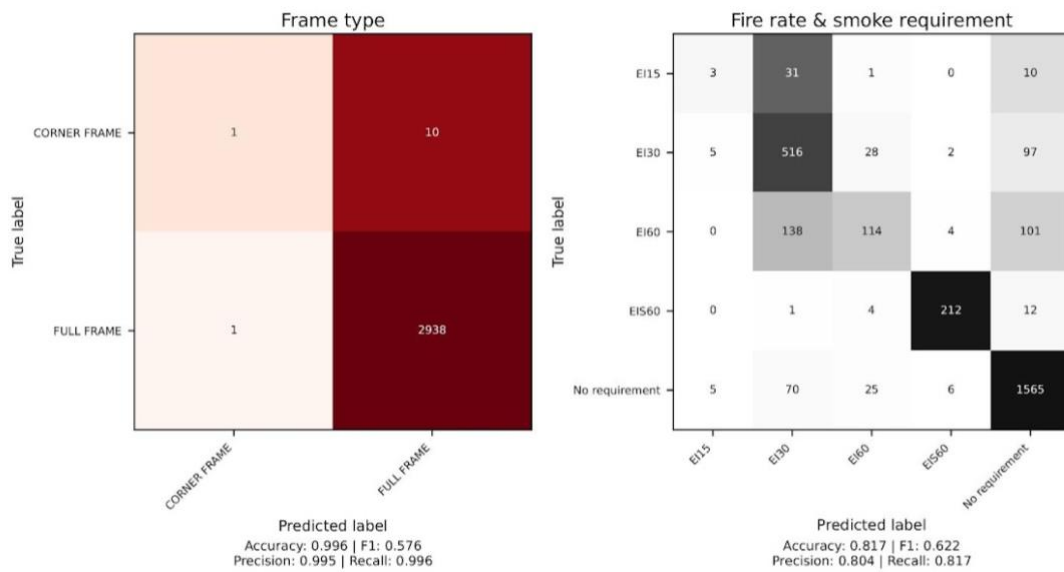


Figure 7: Scenario 1 – confusion matrix results. Cell colors represent row-normalized values (0–1), where darker shades indicate higher prediction accuracy within each class, while the numbers denote raw counts.

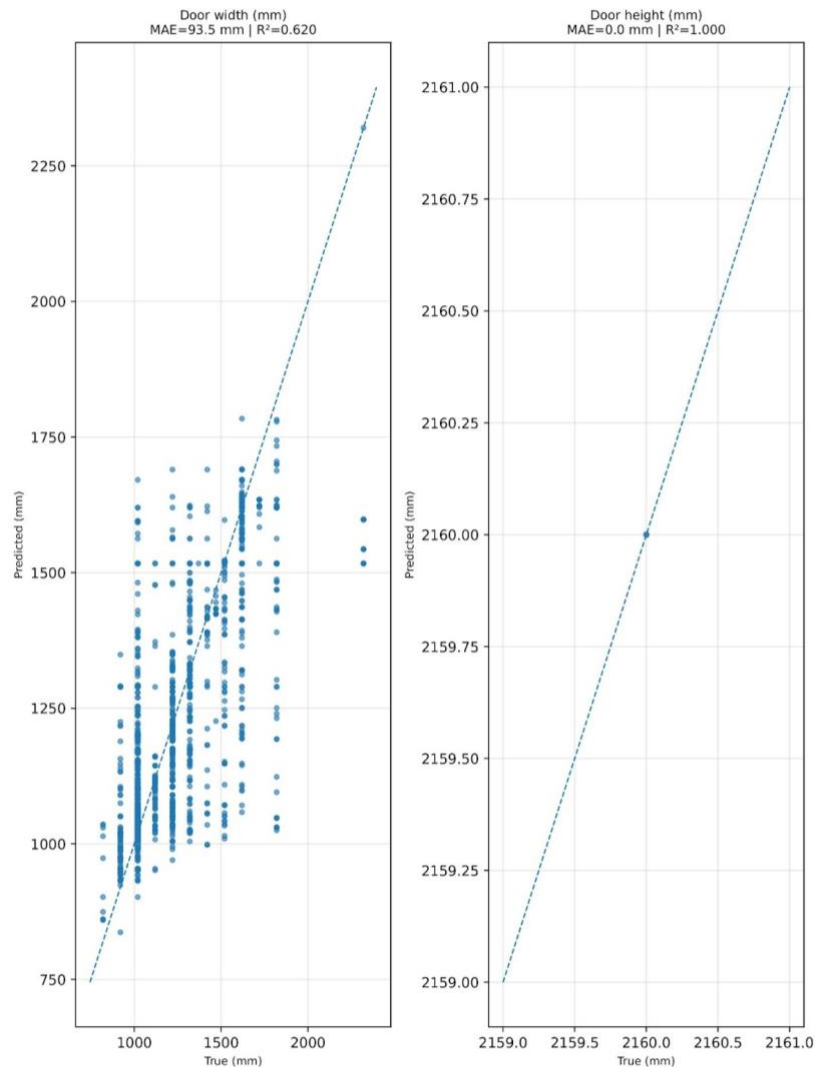


Figure 8: Scenario 1 – regression results.

4.2 Results for Scenario 2: Cross-Project Generalization (Project 1 (hospital) → Project 2 (office))

Moving cross-project from hospital to office sharply exposes domain shift. Relative to S1, Macro-F1 drops for Leaf material (to 0.464 with Accuracy = 0.839; Precision = 0.992; Recall = 0.839). This pattern reflects a majority-class bias driven by class imbalance in the training data and amplified by cross-project distribution shift. In the hospital dataset, leaf material is dominated by steel and wooden categories, whereas the office dataset is predominantly wooden and also includes glass. As a result, the model favors dominant classes while underperforming on less frequent or differently distributed categories, which explains the combination of high precision and low Macro-F1. Macro-F1 also declines for Leaf type (to 0.372 with Accuracy = 0.932; 0.896; 0.932), indicating that high Accuracy can persist under label imbalance while class-wise performance degrades.

Frame type is not effectively estimable in this split because the two projects do not share a sufficiently overlapping label space. A direct comparison shows that one dataset contains only CONCEALED and CORNER FRAME, whereas the other contains only FULL FRAME and CORNER FRAME. Thus, CORNER FRAME is the only shared class, which largely explains why the frame type metrics collapse to zero.

Fire rate and smoke requirement also declines relative to Scenario 1 (Accuracy = 0.724; Macro-F1 = 0.229; 0.697; 0.724) (Figure 9; Figure 10). Regression quality degrades versus S1 for both Door width (MAE = 163.5 mm; $R^2 = 0.037$) and Door height (MAE = 225.8 mm; $R^2 = -1.817$) (Figure 11). $R^2 = -1.817$ indicates that the model

performs worse than a mean-based baseline, highlighting a failure to generalize in this setting. This result is primarily caused by distribution mismatch between the training and test datasets: the hospital dataset contains uniform door height values (2160 mm), whereas the office dataset exhibits a much broader range of values. Consequently, the model learns a constant pattern from the training data and fails to capture the variability in the target dataset. Overall, Scenario 2 shows that cross-project generalization suffers from vocabulary shift, class imbalance, label-set mismatch, and distribution mismatch in continuous variables.

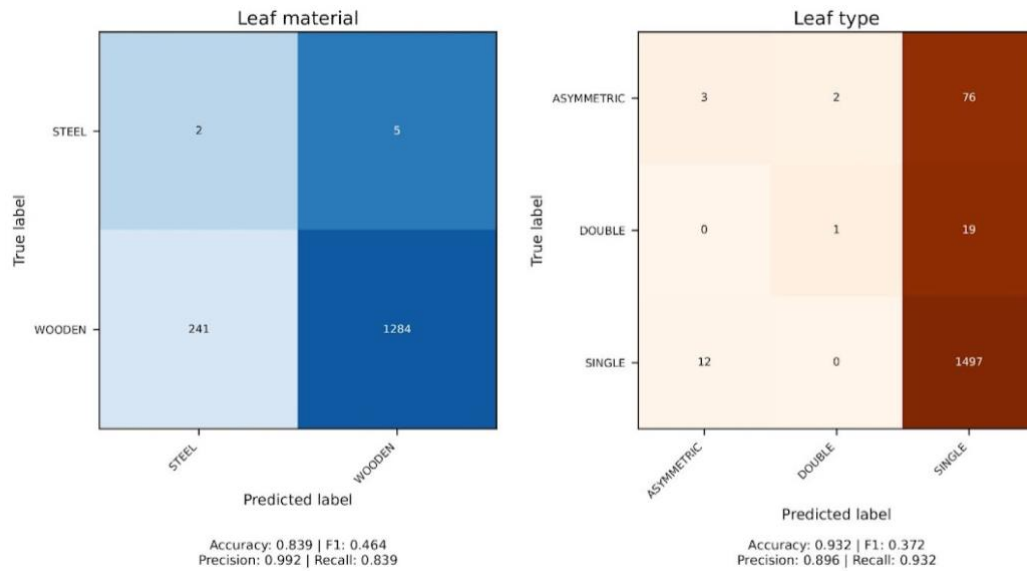


Figure 9: Scenario 2 – confusion matrix results. Cell colors represent row-normalized values (0–1), where darker shades indicate higher prediction accuracy within each class, while the numbers denote row counts.

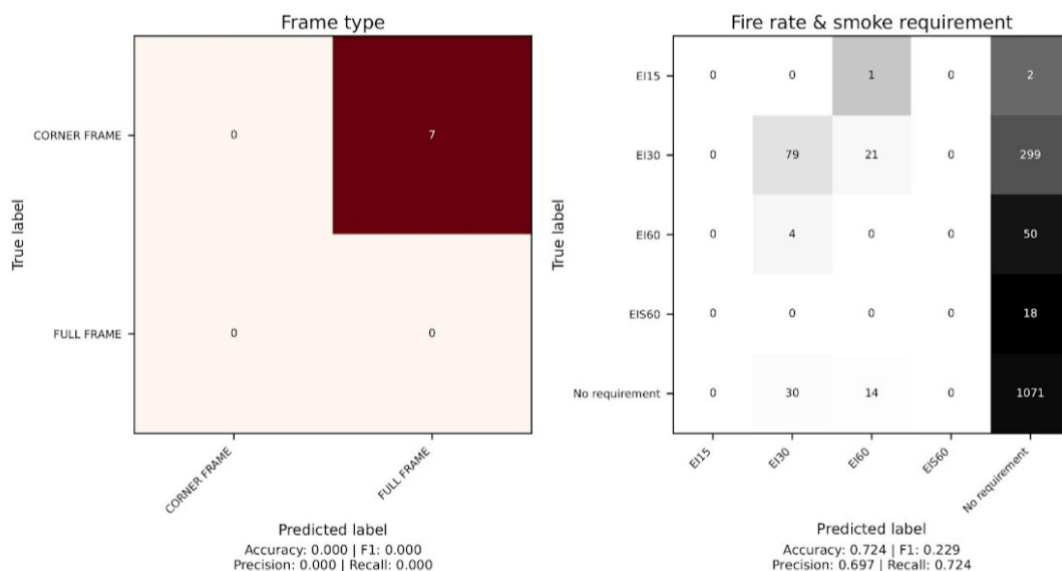


Figure 10: Scenario 2 – confusion matrix results. Cell colors represent row-normalized values (0–1), where darker shades indicate higher prediction accuracy within each class, while the numbers denote row counts.

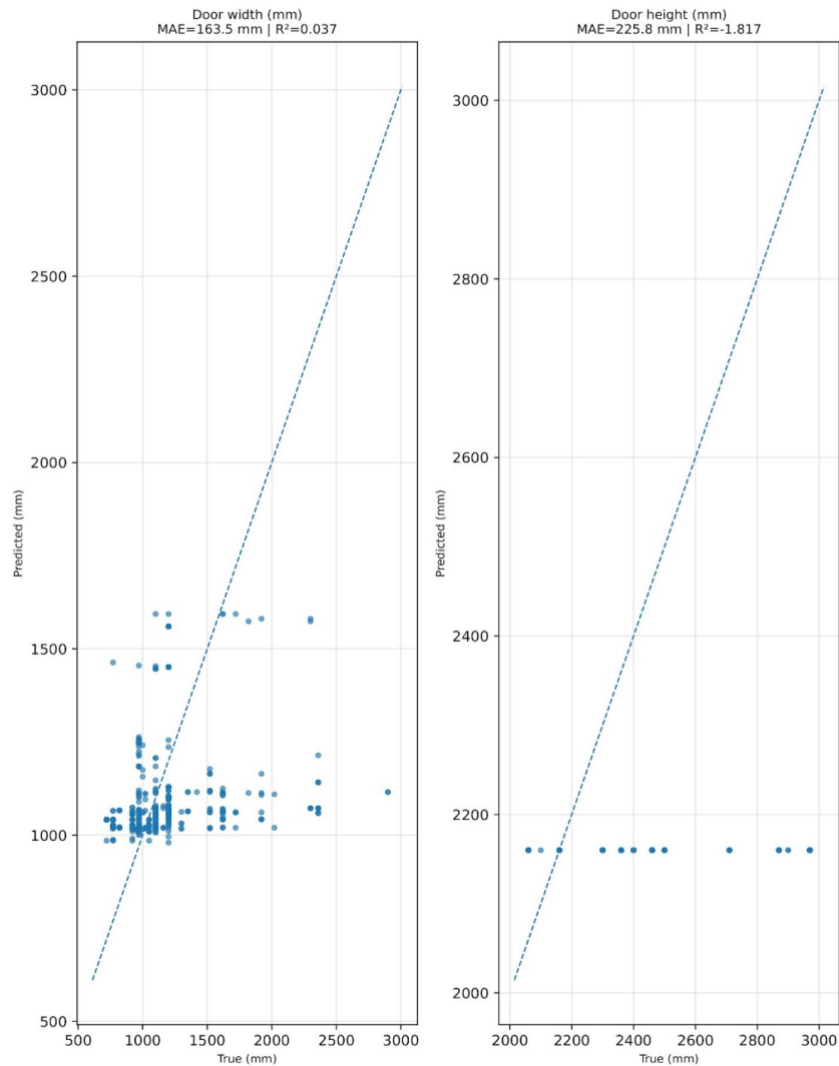


Figure 11: Scenario 2 – regression results.

4.3 Results for Scenario 3: Combined Project Generalization (Project 1 (hospital) + Project 2 (office) → Project 3 (recreational facility))

Training on the hospital and office projects and testing on the recreational facility project does not, by itself, restore Scenario 1-level performance. Relative to Scenario 2, Leaf material weakens further (Accuracy = 0.342; Macro-F1 = 0.209; Precision = 0.918; Recall = 0.342). This decline, despite the larger training set, reflects increased heterogeneity in the source data. While the hospital and recreational datasets are both largely dominated by steel, the office dataset is predominantly wooden and also includes glass, introducing a distribution that is less aligned with the target project. As a result, the larger training set does not provide a more coherent learning signal and instead weakens the consistency of the learned class boundaries. Leaf type recovers modestly (Accuracy = 0.676; Macro-F1 = 0.404; Precision = 0.666; Recall = 0.676) yet remains well below Scenario 1. Frame type cannot be evaluated due to label incompatibilities and limited category overlap in this split. Fire rate and smoke requirement stays low (Accuracy = 0.482; Macro-F1 = 0.262; Precision = 0.460; Recall = 0.482) (Figure 12; Figure 13). Regression further deteriorates compared to Scenario 2: Door width yields MAE = 266.1 mm and $R^2 = -0.134$, while Door height yields MAE = 396.5 mm and $R^2 = -0.809$ (Figure 14). These negative R^2 values indicate that the models perform worse than mean-based baselines, reflecting a failure to generalize under cross-project distribution mismatch. Overall, Scenario 3 shows that enlarging the source domain without target-specific adaptation does not guarantee better generalization; differences in label semantics, class distributions, and continuous-variable patterns remain misaligned with the target project.

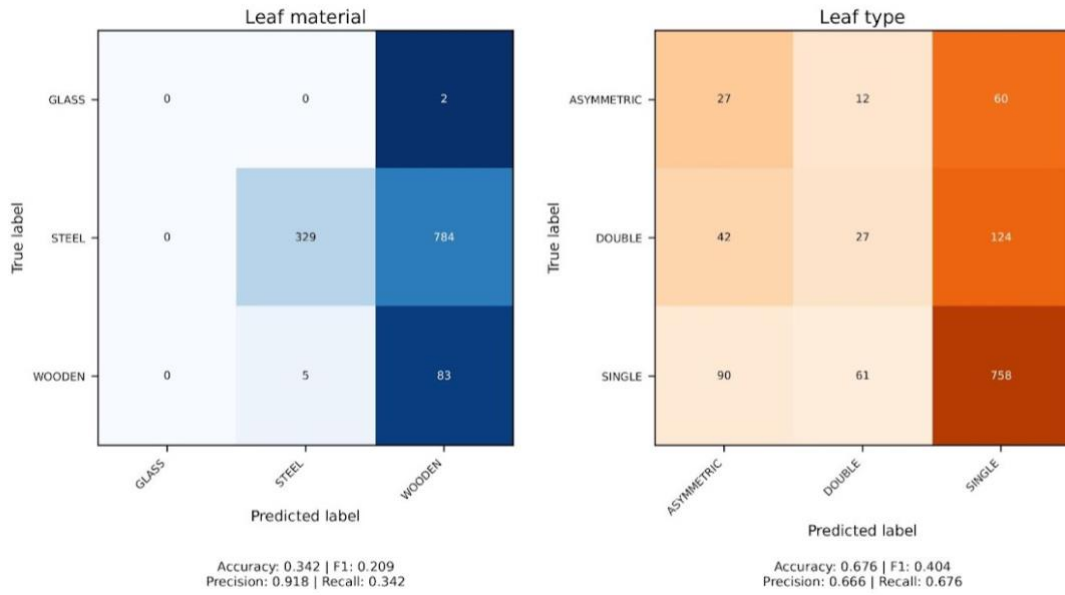


Figure 12: Scenario 3 – confusion matrix results. Cell colors represent row-normalized values (0–1), where darker shades indicate higher prediction accuracy within each class, while the numbers denote raw counts.

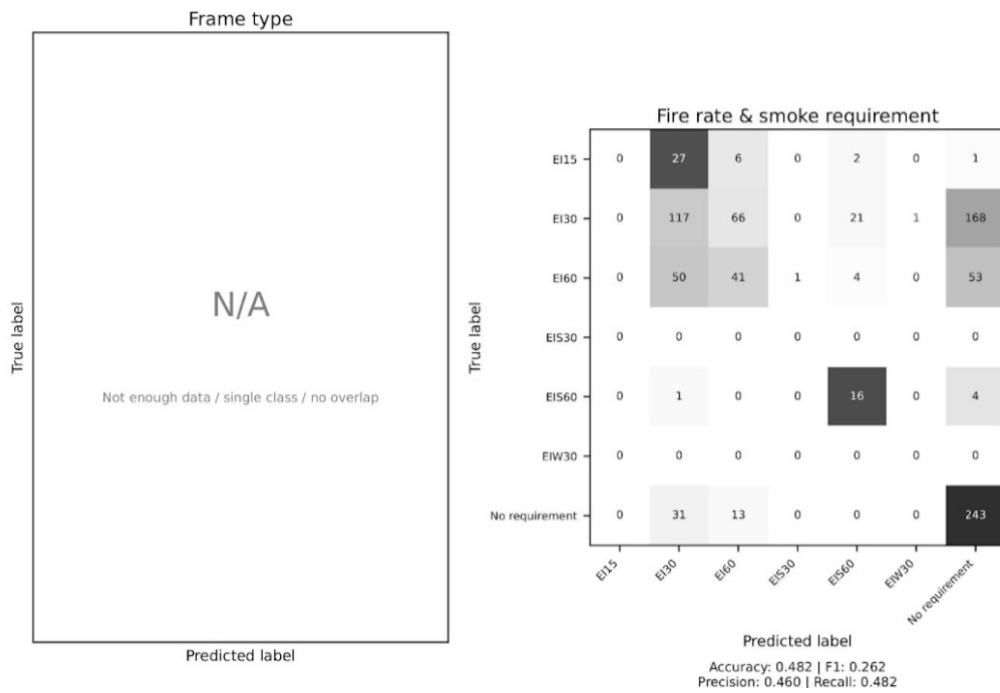


Figure 13: Scenario 3 – confusion matrix results. Cell colors represent row-normalized values (0–1), where darker shades indicate higher prediction accuracy within each class, while the numbers denote raw counts.

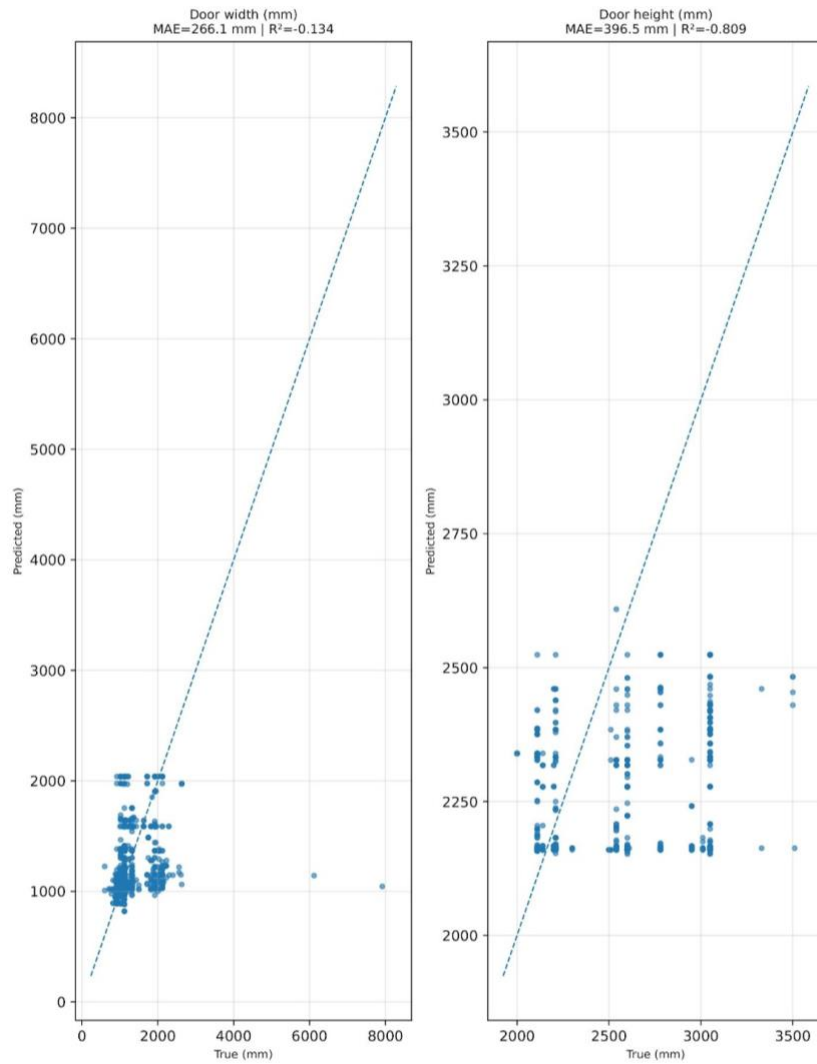


Figure 14: Scenario 3 – regression results.

4.4 Results for Scenario 4: Domain-Augmented Retraining (Project 1 (hospital) + Project 2 (office) + partial Project 3 (recreational facility) → Project 3 (recreational facility))

Adding a small, grouped sample ($\approx 20\%$) from the recreational facility project to fine-tune the model and testing on the held-out 80% yields partial, task-specific improvements over Scenario 3. Leaf material (Accuracy = 0.628; Macro-F1 = 0.336; Precision = 0.903; Recall = 0.628) and Leaf type (Accuracy = 0.706; Macro-F1 = 0.255; Precision = 0.628; Recall = 0.706) improve modestly versus Scenario 3, though they remain far from Scenario 1's within-project levels, evidence that limited target exposure helps but does not fully resolve cross-project drift. Frame type becomes evaluable yet remains weak under imbalance (Accuracy = 0.355; Macro-F1 = 0.090; Precision = 0.929; Recall = 0.355). Fire rate and smoke requirement does not improve and shows a slight decrease compared to Scenario 3 (Accuracy = 0.451 vs. 0.482), remaining within a similarly low performance band (Figure 15; Figure 16). Regression shows the clearest benefit of domain-augmented retraining: Door width improves from Scenario 3 (MAE = 247.7 mm; $R^2 = 0.064$), and Door height also improves (MAE = 330.0 mm; $R^2 = -0.327$), though heights remain poorly explained (Figure 17). Scenario 4 exhibits some recovery over Scenario 3 in regression tasks; however, categorical results remain weak, indicating that light fine-tuning alone does not resolve cross-project inconsistencies.

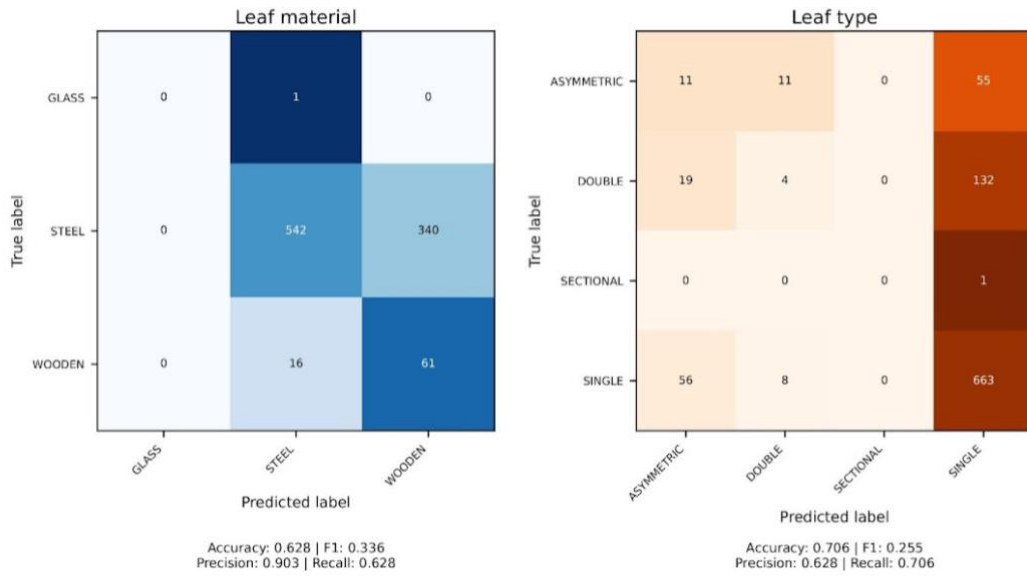


Figure 15: Scenario 4 – confusion matrix results. Cell colors represent row-normalized values (0–1), where darker shades indicate higher prediction accuracy within each class, while the numbers denote raw counts.

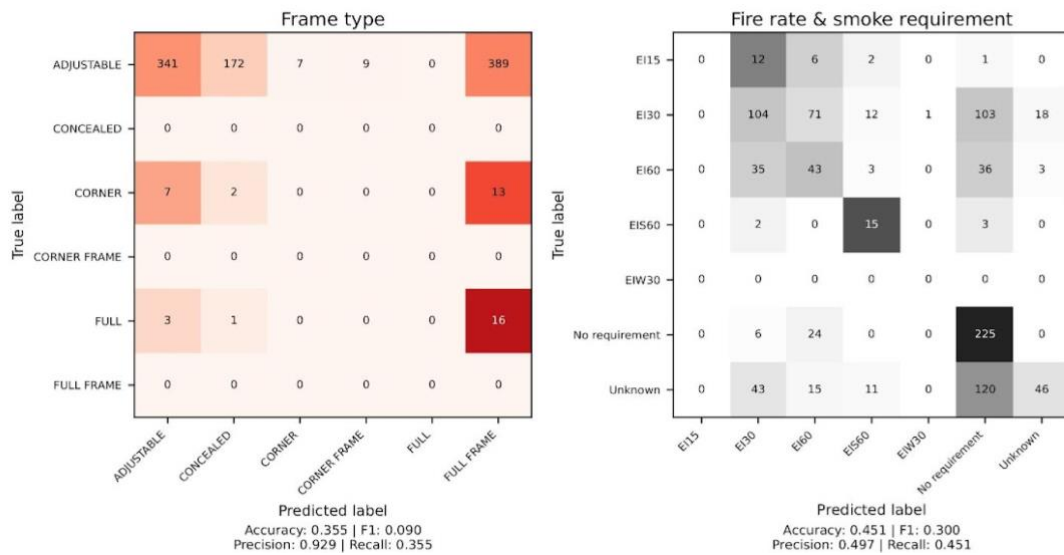


Figure 16: Scenario 4 – confusion matrix results. Cell colors represent row-normalized values (0–1), where darker shades indicate higher prediction accuracy within each class, while the numbers denote raw counts.

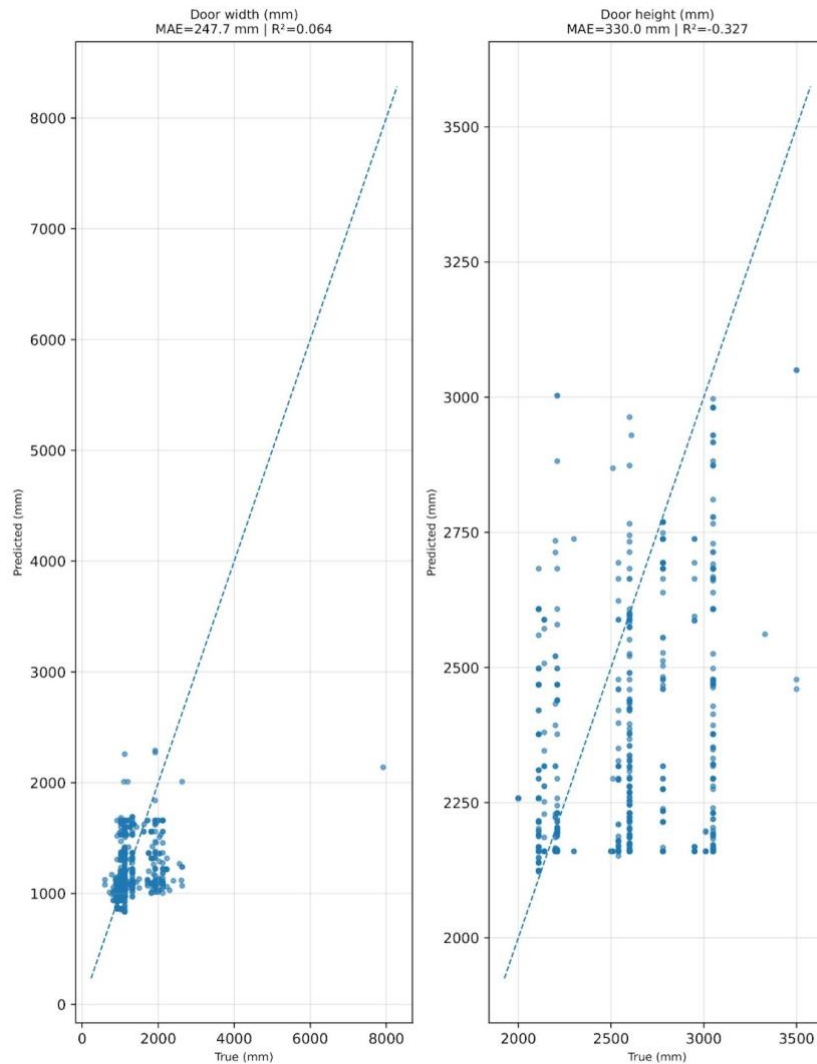


Figure 17: Scenario 4 – regression results.

4.5 Findings

Scenario 1 (hospital to hospital) represents the practical upper bound under matched train/test conditions, as consistent naming, spatial logic, and design standards yield high accuracies and stable macro-F1 values, with residual limitations driven mainly by class imbalance. Scenario 2 (hospital to office) exposes ontology and vocabulary shift across projects. Although overall accuracy may remain deceptively high under class imbalance, macro-F1 and recall decline sharply, and some targets become non-evaluable because of non-overlapping labels. Scenario 3 (hospital + office to recreational facility) demonstrates that simply pooling multiple source projects does not resolve these gaps. While broader training reduces variance, it leaves core semantic and distributional mismatches largely intact, and regression performance deteriorates further. Scenario 4 ((hospital + office + 20% recreational) to 80% recreational, Domain-Augmented Retraining) introduces light target adaptation and yields partial, task-specific improvements, most visibly for low-recall classes and in regression. However, it still falls short of the within-project reliability observed in Scenario 1, indicating that limited target exposure helps but cannot by itself reconcile project-specific conventions.

These trends are reflected in the evaluation metrics (Figure 18, Figure 19). For classification, macro-F1 scores (Figure 18) show a steep drop from Scenario 1 to Scenario 2, with some targets (e.g., frame type) becoming unevaluable, followed by partial recovery under domain-augmented retraining in Scenario 4. For regression, Scenario 1 achieves low mean absolute error (MAE) and high R^2 (Figure 19), but performance collapses in cross-

project settings and only partially recovers with domain-augmented retraining, highlighting the strong dependence of continuous predictions on project-specific conventions. Baseline comparisons further confirm this effect, with majority-class accuracies exceeding 0.90 in some targets, indicating that overall accuracy alone can be misleading under class imbalance.

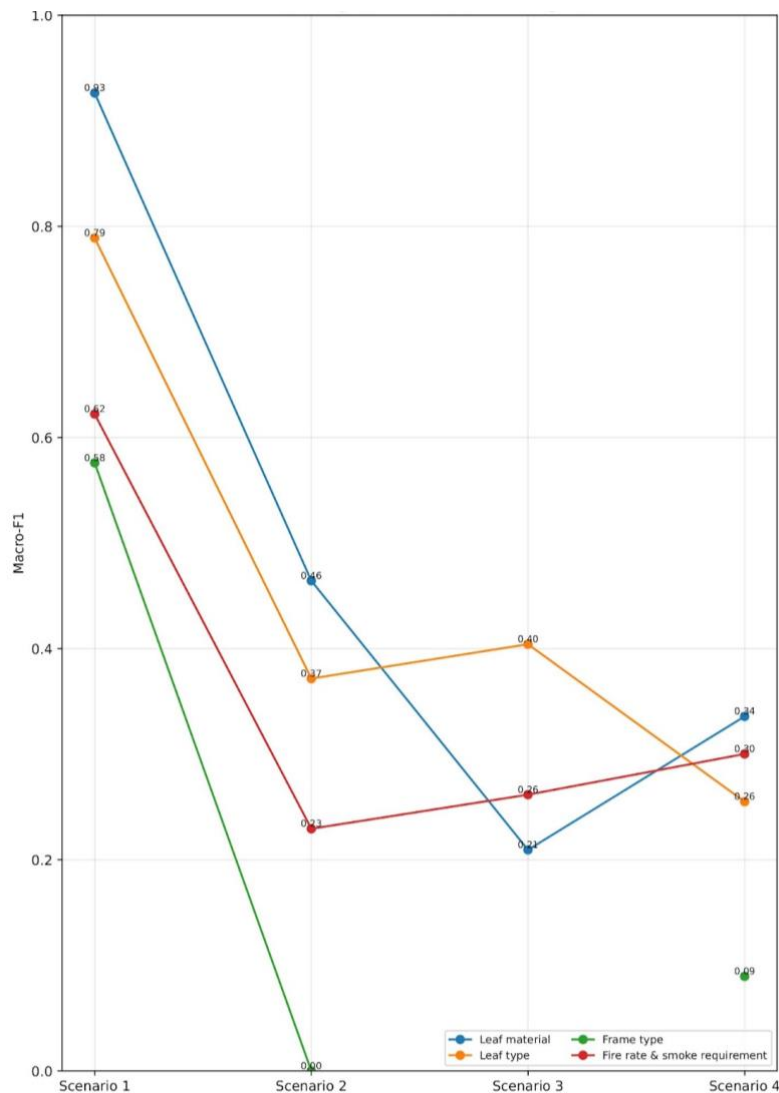


Figure 18: Macro-F1 scores for Leaf Material, Leaf Type, Frame Type, and Fire Rate & Smoke Requirement across scenarios.

Methodologically, two paths are indicated. For immediate design support, models trained and tested within the same project provide reliable results when local data are available. For cross-project use, three elements are essential: (1) consistent label definitions and naming across datasets; (2) explicit handling of class imbalance so rare categories are learned; and (3) light target adaptation, such as fine-tuning on a small sample from the new project, rather than pooled training alone. In practice, combining domain-augmented retraining with clear label mapping and basic data standards offers a practical route to balance generalization with project-specific conventions in BIM-based tools.

These methodological observations can be directly related to common BIM data management practices. Ensuring consistent label definitions corresponds to standardizing parameter naming across projects, for example, through Revit shared parameters or IFC property sets. Class imbalance reflects the uneven distribution of element types within and across projects, which should be considered during data preparation. Similarly, light target adaptation can be interpreted as incorporating a small amount of project-specific data when deploying models in a new BIM

context. Framing these steps within existing BIM data conventions helps bridge the gap between ML workflows and practical implementation in design environments.

From a practical perspective, this study can be seen as an initial step toward integrating predictive models into BIM authoring environments as decision-support tools. Rather than directly predicting full door families, the focus here is on understanding how consistently individual door attributes can be learned and transferred across projects. This is important, as reliable family-level suggestions depend on the stability of these underlying parameters. In future work, such models could be embedded into design workflows, such as a Revit plugin, Dynamo script, or external API, to provide context-aware recommendations during the design process.

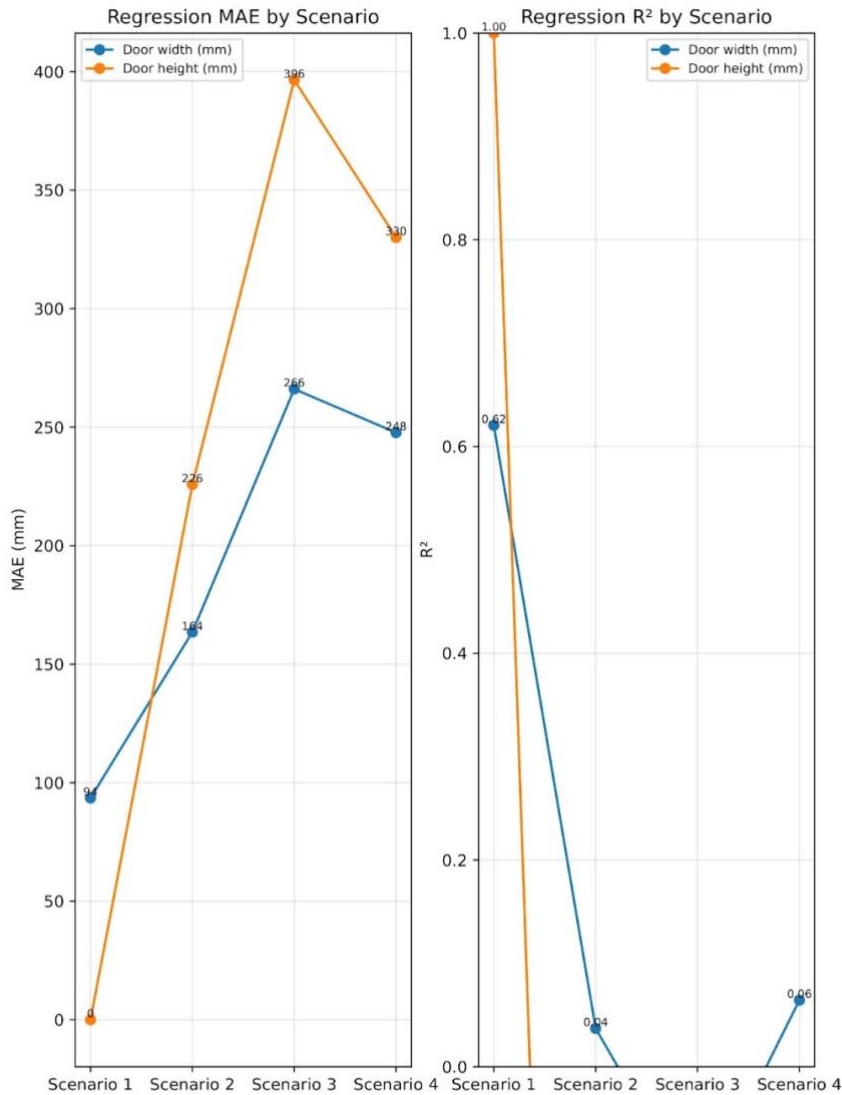


Figure 19: Regression MAE by scenarios.

These observations can be directly interpreted in relation to the data distribution characteristics presented earlier in Section 3.3. As shown in Figures 2–5, the datasets exhibit significant class imbalance and limited overlap in label space across projects. In particular, the dominance of specific categories within individual projects and the absence of certain labels in others create a fragmented learning space for the model.

This structural inconsistency explains the observed performance degradation in cross-project scenarios. Models trained on distributions dominated by a limited set of categories tend to overfit these patterns, leading to reduced recall and macro-F1 when evaluated on datasets with different distributions. Similarly, the presence of non-overlapping labels prevents meaningful generalization for certain targets, as observed in frame type predictions. Therefore, the results should be interpreted not only as a reflection of model limitations but also as a consequence

of underlying data heterogeneity. This highlights the importance of dataset alignment, label standardization, and balanced representation when developing transferable machine learning models in BIM contexts.

5. CONCLUSION

This study investigated the potential of ML to support architectural decision-making by predicting door family specifications from spatial-semantic patterns in BIM models. Two research questions were asked: (1) Can information embedded in past BIM projects be systematically transformed into a decision support mechanism based on predictive modeling for estimating component-level specifications in new projects? (2) Can spatial-functional adjacency between rooms serve as a reliable basis for predicting component family configurations in new design contexts? The results indicate that the proposed approach is effective within single-project settings, where consistent naming conventions and spatial patterns enable reliable predictions. Under domain-augmented retraining, partial improvements are observed, particularly for certain targets and regression tasks. However, the overall performance for several categorical variables remains limited, and does not yet support fully reliable decision-making across projects. Therefore, while the first research question is supported within controlled, project-specific contexts, the second proposition is only partially validated. Spatial-functional adjacency provides a useful but insufficient basis for predicting component configurations in cross-project settings without additional data standardization and adaptation strategies.

Despite these promising results, the study faced several limitations. Cross-project prediction without adaptation resulted in low recall and poor generalization for most parameters, primarily due to inconsistent naming conventions, imbalanced class distributions, and divergent modeling standards between projects. Additionally, some features (such as frame type and fire/smoke rating) were difficult to predict due to their variability and strong dependence on project-specific design conventions. Addressing these issues will require larger and more standardized datasets, domain adaptation techniques, and semantic or hierarchical encoding strategies to improve generalizability and extend applicability to other building components in future work.

From a practical perspective, this study presents a proof of concept for a predictive modeling approach that can inform future decision support systems, rather than a fully implemented DSS. The current framework operates as an analytical pipeline and does not include interactive or user-driven components. However, it demonstrates how BIM-derived data can be utilized to generate predictive insights that may support design decisions in future, more integrated implementations. Such systems have the potential to reduce manual effort, improve consistency, and support more efficient documentation processes; however, these benefits are not directly evaluated in this study and should be considered as intended outcomes that require validation through future implementation and user-based assessment.

DATA AVAILABILITY

The research data of this study were obtained from ENKA Construction through its ENKA Design Center. Due to confidentiality agreements and company policies, these data are not publicly available. All data were anonymized prior to analysis, and no project-specific or client-identifiable information is disclosed. The study uses only structured BIM metadata related to spatial configurations and component attributes. The code used for data preprocessing and model training is available from the corresponding author upon reasonable request.

REFERENCES

- Ackoff, R. L. (1989). From data to wisdom. *Journal of Applied Systems Analysis*, 16, 3–9.
- Azhar, S. (2011). Building information modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry. *Leadership and Management in Engineering*, 11(3), 241–252. [https://doi.org/10.1061/\(ASCE\)LM.1943-5630.0000127](https://doi.org/10.1061/(ASCE)LM.1943-5630.0000127)
- Becerik-Gerber, B., & Kensek, K. (2010). Building information modeling in architecture, engineering, and construction: Emerging research directions and trends. *Journal of Professional Issues in Engineering Education and Practice*, 136(3), 139–147. [https://doi.org/10.1061/\(ASCE\)EI.1943-5541.0000023](https://doi.org/10.1061/(ASCE)EI.1943-5541.0000023)



- Bilal, M., Oyedele, L. O., Qadir, J., Munir, K., Ajayi, S. O., Akinade, O. O., Owolabi, H. A., Alaka, H. A., & Pasha, M. (2016). Big data in the construction industry: A review of present status, opportunities, and future trends. *Advanced Engineering Informatics*, 30(3), 500–521. <https://doi.org/10.1016/j.aei.2016.07.001>
- Boje, C., Guerriero, A., Kubicki, S., & Rezgui, Y. (2020). Towards a semantic construction digital twin: Directions for future research. *Automation in Construction*, 114, 103179. <https://doi.org/10.1016/j.autcon.2020.103179>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- Bryde, D., Broquetas, M., & Volm, J. M. (2013). The project benefits of building information modelling (BIM). *International Journal of Project Management*, 31(7), 971–980. <https://doi.org/10.1016/j.ijproman.2012.12.001>
- Eastman, C., Teicholz, P., Sacks, R., & Liston, K. (2011). *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors (2nd ed.)*. Wiley.
- Gero, J. S. (1990). Design prototypes: A knowledge representation schema for design. *AI Magazine*, 11(4), 26–36. <https://doi.org/10.1609/aimag.v11i4.854>
- Guo, P., Xue, H., Ma, J., & Cheng, J. C. P. (2025). Advancing BIM information retrieval with an LLM-based query-domain-specific language and library code function alignment system. *Automation in Construction*, 178, 106374. <https://doi.org/10.1016/j.autcon.2025.106374>
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction (2nd ed.)*. Springer.
- International Organization for Standardization. (2025). *ISO 16739-1: Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries*. <https://www.iso.org/>
- Jin, S., Kim, D., Lee, J., & Lee, D. (2026). Evaluating large language models (LLMs) for semantic interpretation of IFC-based BIM data. *Automation in Construction*, 185, 106879. <https://doi.org/10.1016/j.autcon.2026.106879>
- Jung, N., & Lee, G. (2019). Automated classification of building information modeling (BIM) case studies by BIM use based on natural language processing (NLP) and unsupervised learning. *Advanced Engineering Informatics*, 41, 100917. <https://doi.org/10.1016/j.aei.2019.04.007>
- Jurafsky, D., & Martin, J. H. (2023). *Speech and language processing (3rd ed.)*. <https://web.stanford.edu/~jurafsky/slp3/>
- Kensek, K. (2014). Integration of environmental sensors with BIM: Case studies using Arduino, Dynamo, and the Revit API. *Informes de la Construcción*, 66(533), e044. <https://doi.org/10.3989/ic.13.151>
- Khallaf, R., & Khallaf, M. (2021). Classification and analysis of deep learning applications in construction: A systematic literature review. *Automation in Construction*, 129, 103760. <https://doi.org/10.1016/j.autcon.2021.103760>
- Krygiel, E., & Nies, B. (2008). *Green BIM: Successful sustainable design with building information modeling*. Wiley.
- Magnavaca de Paula, L., Oloufa, A., & Tatari, O. (2026). Integrating BIM and machine learning for energy and carbon performance prediction in office building design. *Eng*, 7(2), 73. <https://doi.org/10.3390/eng7020073>
- Marcher, C., Giusti, A., & Matt, D. T. (2020). Decision support in building construction: A systematic review of methods and application areas. *Buildings*, 10(10), 170. <https://doi.org/10.3390/buildings10100170>
- Oesterreich, T. D., & Teuteberg, F. (2016). Understanding the implications of digitisation and automation in the context of Industry 4.0. *Computers in Industry*, 83, 121–139. <https://doi.org/10.1016/j.compind.2016.09.006>
- Oxman, R. (2017). Thinking difference: Theories and models of parametric design thinking. *Design Studies*, 52, 4–39. <https://doi.org/10.1016/j.destud.2017.06.001>

- Park, I., Kim, Y., Kim, K., & Chin, S. (2026). BIM–LLM integration in AECO workflows: Applications, validation, and future directions. *Automation in Construction*, 187, 106905. <https://doi.org/10.1016/j.autcon.2026.106905>
- Pauwels, P., Zhang, S., & Lee, Y.-C. (2017). Semantic web technologies in AEC industry: A literature overview. *Automation in Construction*, 73, 145–165. <https://doi.org/10.1016/j.autcon.2016.10.003>
- Power, D. J. (2002). *Decision support systems: Concepts and resources for managers*. Greenwood/Quorum Books.
- Solihin, W., & Eastman, C. (2015). Classification of rules for automated BIM rule checking development. *Automation in Construction*, 53, 69–82. <https://doi.org/10.1016/j.autcon.2015.03.003>
- Succar, B. (2009). Building information modelling framework: A research and delivery foundation for industry stakeholders. *Automation in Construction*, 18(3), 357–375. <https://doi.org/10.1016/j.autcon.2008.10.003>
- Volk, R., Stengel, J., & Schultmann, F. (2014). Building information modeling (BIM) for existing buildings—Literature review and future needs. *Automation in Construction*, 38, 109–127. <https://doi.org/10.1016/j.autcon.2013.10.023>
- Watson, A. S. (2011). Digital buildings—Challenges and opportunities. *Advanced Engineering Informatics*, 25(4), 573–581. <https://doi.org/10.1016/j.aei.2011.07.003>
- Yin, M., Tang, L., Webster, C., Li, J., Li, H., Wu, Z., & Cheng, R. C. K. (2023). Two-stage text-to-BIMQL semantic parsing for building information model extraction using graph neural networks. *Automation in Construction*, 152, 104902. <https://doi.org/10.1016/j.autcon.2023.104902>
- Zhang, L., & Issa, R. R. A. (2013). Ontology-based partial building information model extraction. *Journal of Computing in Civil Engineering*, 27(6), 576–584. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000277](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000277)