applied sciences

MDPI

Article

Automation of Learning Workflows for 3D Modeling Skills in Engineering Education

Francisco Salmerón-Medina [1,2,*], María Alcalde [1], Diego Canales [1], Fabio Gómez-Estern [1] and Francisco Valderrama-Gual [2]

[1] Departamento de Ingeniería, Escuela de Ingenieros, Campus Sevilla, Universidad Loyola Andalucía, 41704 Dos Hermanas, Spain; dcanales@uloyola.es (D.C.)

[2] Departamento de Ingeniería Gráfica, Escuela Técnica Superior de Ingeniería (ETSi), Universidad de Sevilla, Camino de los Descubrimientos, s/n, Isla de la Cartuja, 41092 Sevilla, Spain; fvg@us.es

* Correspondence: fsalmeron@uloyola.es

Abstract

This paper presents a novel platform for automated self-paced learning in Computer-Aided Design (CAD) courses within engineering education. The platform fully automates the entire learning cycle, including exercise generation, submission, scheduling, test design, and grading. The central hypothesis posits that complete automation reduces repetitive tasks for instructors, allowing them to dedicate more time to individualized student support. The system also provides key advantages: it generates unique exercises for each student to prevent plagiarism while maintaining comparable complexity; it delivers instantaneous grading and feedback to enhance motivation; and it enables students to work with almost any CAD software, as the evaluation relies on physical properties rather than commercial tools. After several years of successive testing and refinement, the tool can generate and accurately grade frequent activities in large student cohorts, providing abundant data points. Their statistical analysis, via multiple approaches, confirms that the system reliably produces individualized exercises, reduces grading errors, and offers prompt, consistent feedback, thereby supporting a more efficient and engaging learning process.

Keywords: computer aided design; engineering education; automated assessment; internet web-based learning; educational technology; problem-based learning; geometry-based evaluation

1. Introduction

Spatial abilities have been identified as crucial abilities to develop in engineering education [1,2]. The set of available digital tools to support this development is continuously increasing, and now includes virtual reality [3]. Simultaneously, ref. [4] emphasizes the pivotal role of interaction in virtual environments for developing spatial reasoning—a fundamental ability to gain proficiency in CAD modeling tasks. Building on these insights, the present study investigates CAD learning via an automated platform for generating and correcting modeling exercises, aiming to evaluate how such a system supports spatial and design competence development in engineering students.

A fundamental course in introductory technical drawing, encompassing both 2D and 3D aspects, is part of the first year of engineering education worldwide [5]. This subject has undergone significant evolution from paper-based drawing to digital modeling with CAD software like SolidWorks 2025. The evolution has prompted a comprehensive re-evaluation of the teaching methodologies. Rather than focusing on memorization of

Academic Editor: Alexander Barkalov

Received: 6 April 2026
Revised: 29 April 2026
Accepted: 7 May 2026
Published: 13 May 2026

procedural steps and manual ability, the emphasis now lies on creating 3D models using CAD tools, implying a substantial change in the competencies developed, widely analyzed in [6]. The potential for automating the learning process in CAD is enormous, but it has not been deeply analyzed in academic literature.

This study specifically addresses the automation of learning 3D competencies using a web-based tool called Doctus, see [7]. While automated assessment tools have been previously used in many areas [8–10], their application in CAD learning is a novel approach, and it represents a great challenge. In ref. [11], the authors present suitable criteria and methods for the systematic evaluation of exercises based (a) on physical parameters of the generated 3D parts, and (b) on SolidWorks' own comparison operations. However, it does not address the generation of problem statements or the automated evaluation of a group of students. To achieve this, it is necessary to integrate the procedures for generating problem statements and validating parts into a platform that organizes and automates the entire process. This is accomplished in our article using the Doctus platform. Furthermore, the article presents the results of extensive experience with numerous students and analyzes them statistically.

On the other hand, reference [12] proposes an evaluation system based on exploratory information obtained from the figure file submitted by the student: general parameters, characteristics, topology, relationships, and dependencies. This is a powerful approach in that it considers several profound aspects of the piece's design, some of which are not addressed. However, it raises challenges that can complicate the grading process: What exactly does averaging across different approaches mean? How can one avoid favoring a specific construction sequence when several are valid? Finally, the presented application integrates all the evaluation elements but does not automate the process for the entire group of students, neither in generating personalized prompts nor in the systematic evaluation of submissions. For this, the support of a web tool like Doctus is necessary. Lastly, article [12] lacks an analysis of the results from extensive experience working with students.

Moreover, modern tools such as CADuBoost [13] and AMCAD [14] use point cloud comparison, multi-view image analysis, and other tools for geometric evaluation. In CADuBoost, non-geometric evaluation is performed by extracting and analyzing design history and constraint information via the 3D CAD system's API. A comparable approach is found in [15,16], which utilizes the Creo Parametric VBAPI to perform automated evaluations. Unlike the proposed systems, which necessitate the Creo Parametric environment to execute a dual-layer verification: a structural analysis of the feature tree coupled with a validation of physical properties. Again, these tools are in-depth assistants for educators that do not provide 100% automated exercise personalization and evaluation for large student groups. Insight is traded for efficiency, which makes sense in a unique evaluation course; the lack of immediate feedback for students and instructors requires continuous evaluation and active learning environments. The web-based platform Doctus serves to streamline the whole process.

As a major learning outcome in our CAD module, students learn how to model a 3D object using SolidWorks, starting from 2D drawings. To achieve it, they must become proficient in many standard competencies of CAD courses, including 2D drawing, 3D modeling, major CAD software, and CAD-based design projects [6]. In this work, the competencies have been categorized and subdivided into micro-competencies, based on [17,18]. The 3D problem to be solved can be split into different steps using main modelling operations: extrusions, revolutions, sweepings, and multi-section [19], and other adding or removing operations to refine the final 3D model according to the provided task descriptions.

It is necessary to consider that drawing skills and spatial vision are particular competencies that students have barely acquired in pre-university education [20,21], and their related natural skills are very uneven. For this reason, designing a personalized learning path based on practical exercises, with difficulty gradually, is essential [13]. The advantages of automated CAD assessment extend beyond instructor time savings to include immediate feedback [22,23], formative assessment [24,25], a more complex and adapted learning process to each student, and more exercises for practice that can be evaluated. Immediate feedback allows one to continue the learning process at home without waiting for the instructor's correction and allows each student to study for the time they need, which will be very different from one student to another. All these result in a higher probability of success in acquiring 3D competencies [26].

2. Materials and Methods

This section breaks down the methodology into several layers that correspond to descending levels of abstraction. The upper layer, the set of competences, represents the final goal of the methodology. Then, the learning process throughout the course is explained, followed by the principles for building parameterizable exercises within the course. Finally, the underlying technology supporting the life cycle is described as a set of automation tools integrated and coordinated within the Doctus platform.

2.1. Micro-Competences Structure in the Learning Process

The main competencies in CAD subjects include Major CAD Software, 2D Drawings, 3D Modeling, and Design Project [6]. Our work primarily focuses on the 2D Drawing competence (necessary for interpreting and collecting the required information for modeling) and the 3D Modeling competence (used for performing the requested 3D solid modeling). The 3D Modeling competence can be split into micro-competences that facilitate and structure the acquisition of the main one. In this context, operations such as extrusion, revolution, sweeping, or multi-section covering are the basic operations that support modeling, allowing the addition or removal of material. This also includes 'dress-up' operations to optimize the result.

To guarantee the acquisition of each micro-competence individually, the learning path in our university has been designed in the following way:

- The course is divided into 4 units, each one corresponding to one main modeling operation: extrusions, revolutions, sweepings, and multi-section.
- The correct acquisition of skills within each unit is assessed using a sequence of between three and seven steps.
- A prerequisite exercise of low difficulty to test if the main operation has been perfectly learned, providing students with a guide of different steps to follow, as can be seen in Figure 1. A grade of 9 or 10 (out of 10) is necessary to continue to the next step. This exercise contains a small number of operations to facilitate error location if the student does not pass the task easily. It is developed at home to guarantee that each student has the time they need to assimilate the new concept.
- A more complex exercise with moderate difficulty is unlocked if the student gets a minimum grade of 9 in the previous exercise (see Figure 2). In this assignment, without intermediate submissions, the student must prove that she can apply the acquired skill to different contexts. It is also solved at home.
- A final examination after the second and fourth units. In these exercises, students must apply concepts from previous units, demonstrating assimilation correctly under the instructor's vigilance and within a limited time constraint. It is carried out at the university and within a limited time.

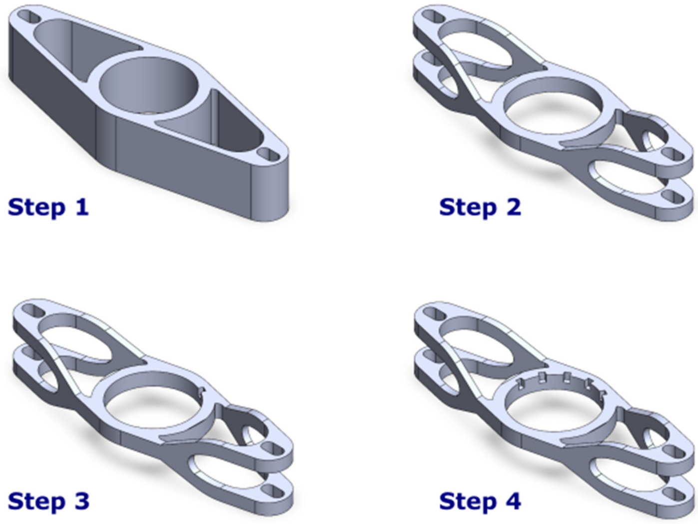

Figure 1. Step by step exercise.

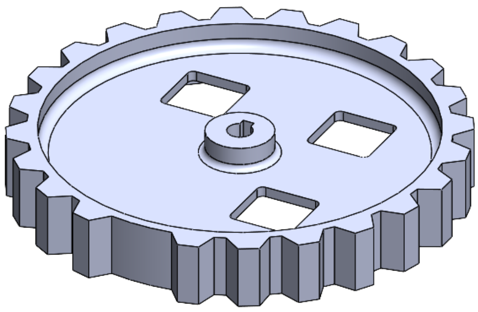

Figure 2. Regular exercise.

A further important characteristic of this learning process design is that the student receives immediate feedback due to the automated correction, providing an immediate boost in motivation. In addition, the exercises can be delivered once a day for two weeks, so the learning process is not interrupted by a low grade; conversely, students are motivated to repeat the exercise to improve their score [18].

This incremental submission model aligns with the scaffolding theory proposed by [27], where a complex task is partitioned into smaller, manageable units. By requiring the validation of a single operation at each step, the system reduces the learner's cognitive frustration and ensures the mastery of foundational modeling techniques before increasing the task's complexity. Following this pedagogical line, recent studies in engineering education [28] have demonstrated that integrating such scaffolding with AI-powered tutoring and simulations can effectively guide students through complex conceptual transitions, presenting a valuable opportunity to enhance the feedback capabilities of platforms like Doctus.

2.2. Parametrization of 3D Models

As a design principle, students should deal with tasks that are different enough to minimize the chances of plagiarism and similar enough to follow the same learning path. The challenge of automatically creating tasks with different descriptions for each student but similar in difficulty level is solved in this paper via parametrization.

To generate personalized exercises, we start with a master 3D model, from which we can generate any number of modified versions or variations. Each variation is obtained by changing the descriptive dimensions of the original model. In this application, the descriptive dimensions of a solid body are lengths, diameters, angles, and other numerical values that ultimately determine the 3D shape and its interior relations. For instance, in

Figure 3, the solid shape is determined by the values of parameters C1, C2, C3, C4, and C5. To generate different values for these, Doctus must produce a particular set of 4 numbers for each student, based on their ID digits.

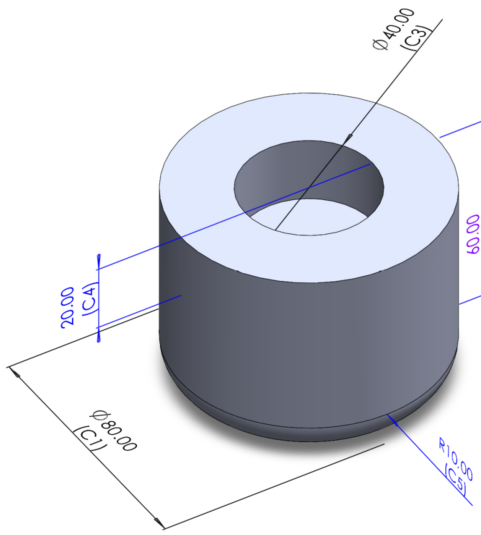

Figure 3. Sample part.

However, not every set of parameters will be valid to maintain the consistency and functionality of the generated 3D shape. In the example, it is clear that C1 > C3 is a necessary condition. A way of imposing such constraints when generating parameter values for each student is done in four steps:

In the first step, the former parameters will be called "external", while the latter parameters are called "internal". "Externality" is a relative term that helps the instructor see the geometric sense of parameter dependence. In general, external dimensions are freely chosen first (e.g., C1, Figure 3), and the rest, the internal ones, depend on the external. In Table 1, an ordered parameter list is [C1, C2, C3, C4, C5].

Table 1. Ordered parameter list.

Parameter	Units	Dependence	Type	Interval Start (Cs)	Interval End (Ce)
C1	mm	-	Interval	10	100
C2	mm	C1	Interval	0.6 × C1	0.8 × C1
C3	mm	C2	Interval	0.5 × C2	0.7 × C2
C4	mm	C3	Interval	0.4 × C3	0.6 × C3
C5	mm	C2	Single value	C2/6	-

The first and second steps require a proper understanding of the master 3D model. For instance, the diameter of a hole in a plate must be defined as a part of the length of the plate to avoid impossible cases, such as a hole bigger than the plate itself. This step might be cumbersome at times, but master models are redesigned, at most, once every academic year.

The third step continues iterating through the remaining parameters, assigning either specific values or ranges based on linear functions of the preceding parameters. Finally, in the last step, once all parameter intervals are determined, they serve as inputs for a shape generation algorithm. This algorithm utilizes the student's ID to further personalize the output by selecting specific points within the defined parameter ranges, ensuring a unique result for each student.

The sequential process for defining parameters is summarized in Figure 4. Table 1, on the other hand, shows how this process is adapted to the example in Figure 3.

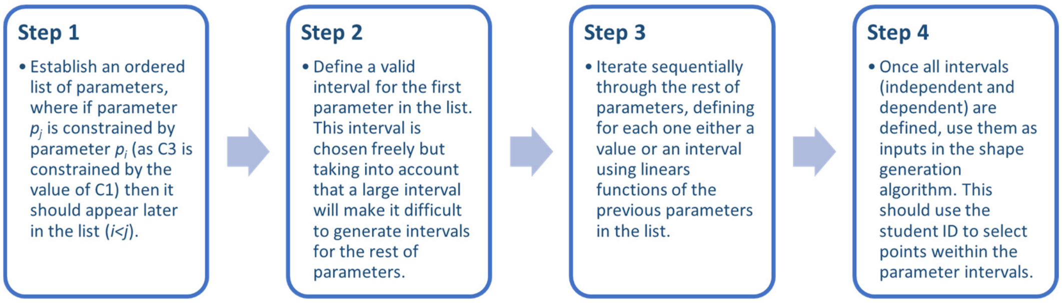

Figure 4. Process parameter definition.

Specifically, the student's identification number consists of eight digits, which the system processes in pairs (e.g., AB-CD-EF-GH). The variable AB represents a specific two-digit integer (ranging from 00 to 99) extracted from one of these pairs. This value serves as the seed for the linear mapping within the parameter interval [Csi, Cei], calculated as follows:

$$\mathrm{Ci} = \mathrm{Csi} + (\mathrm{Cei} - \mathrm{Csi})/99 \cdot \mathrm{AB} \quad (1)$$

By assigning different pairs (CD, EF, etc.) to different parameters (C2, C3...), the system ensures that the resulting 3D model is a unique instantiation for each student.

By using this procedure, Doctus will be able to select valid values for the descriptive dimensions for every student, based on their ID numbers. The students will be presented with significantly different instantiations of the 3D model. Instantiations will differ from one another in several descriptive parameters; hence, they will not be related by simple scaling/translating operations. This reduces the risk of plagiarism.

This risk can be further mitigated if we not only reparametrize the dimensions but also suppress or select (switch) some of the 3D operations to be performed in the sequence to reach the final model. This is possible without sacrificing the homogeneous difficulty requirement if we offer different alternatives in minor steps of the 3D design sequence that do not affect further operations.

The operational switching procedure consists of selecting—for each student—between analogous geometric features at specific stages of the 3D design process. These variations include substituting chamfers for filets or alternating between different pattern types and instance counts. Such operations are dynamically toggled based on the student's unique identifier (ID) via a custom Visual Basic that interfaces with the SolidWorks API to manage part configurations. This mechanism enables the programmatic generation of distinct model variants from a single master file. Furthermore, these operations are considered equivalent in terms of cognitive load and technical difficulty, as they reside within the same feature-based modeling hierarchies of most CAD environments. Representative examples, such as linear versus circular patterns or variations in instance density, are illustrated in Figure 5.

The operation switching for each student is made again as a function of the student ID number digits, as in the descriptive dimension procedure. The implementation details are omitted for brevity.

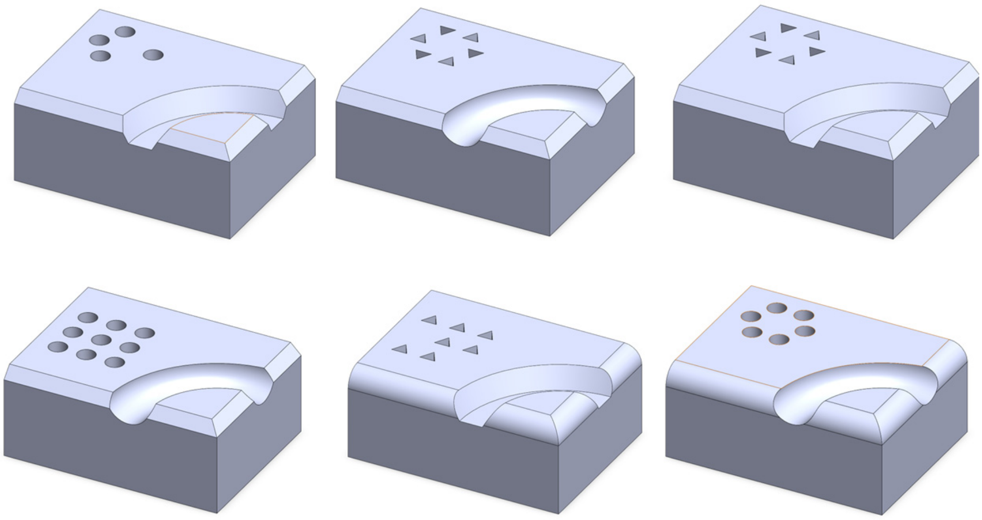

Figure 5. Differentiation through similar operations.

One example of model variation with switching operations is shown in Figure 5, where some discrete variations have been introduced in the 3D part design sequence. This feature discourages plagiarism as the differences become insurmountable. A common resource for plagiarism is sharing video tutorials that show how to solve one of the parametrized variations. This option is therefore rendered ineffective by switching operations.

2.3. Automation of the Learning Process

After parameterization and operation switching, task descriptions are generated as a function of student ID. As illustrated in Figure 6, the system architecture follows a continuous data flow: (i) input: unique 2D problem statements and master solutions are automatically generated from a 3D master model using Visual Basic and SolidWorks 2025 APIs based on student IDs; (ii) processing: students complete the modeling tasks and submit their files to the Doctus platform; and (iii) output: a MatLab R2025b -based engine extracts the physical properties, compares them against the database, and provides instantaneous feedback and grading. This automated process is implemented in the following way.

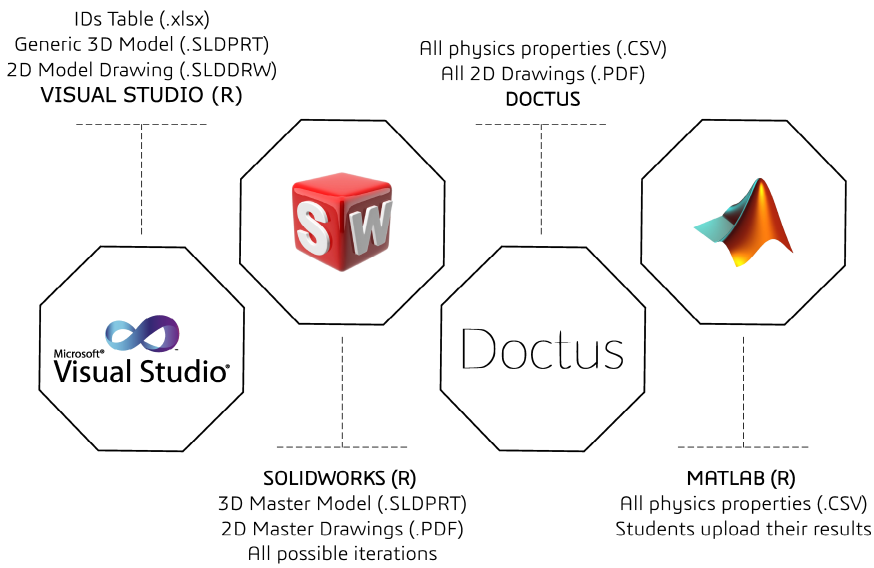

Figure 6. Visual Basic–Solidworks–Doctus–Matlab architecture.

1. Create a data table containing student IDs.
2. Parametrize the master 3D model in SolidWorks.
3. Automatically generate personalized 2D model drawings for individual tasks in SolidWorks.
4. Visual Basic, through a call to SolidWorks under the Visual Basic for Applications automation framework, generates for each student ID:
    a. A table with the physical properties (.csv):
        i. (X, Y, Z) coordinates of the center of mass.
        ii. Mass.
        iii. Volume.
        iv. Perimeter surface.
    b. a personalized 2D model drawing in PDF.
5. This information is uploaded to Doctus, the platform for automated correction, more specifically:
    a. The table with the physical properties.
    b. The personalized 2D model drawing in PDF.
    c. Students' tasks can be subdivided into:
    d. Download personalized 2D model drawings in PDF.
    e. Execute the SolidWorks 3D model.
    f. Obtain the physical parameters of its 3D personalized model using SolidWorks commands. An example is shown in Figure 7.
    g. Submit their numerical results of the physical properties to Doctus through a text template (with Matlab R2025b pieces of code) preloaded in Doctus (Figure 8).

```
Mass properties

Density = 0.00 grams per cubic millimeter

Mass = 404.87 grams

Volume = 404,873.19 cubic millimeters

Surface area = 36,989.69  square millimeters

Center of mass: ( millimeters )
 X =−28.60
 Y =−29.75
 Z =−4.68
```

Figure 7. Mass properties.

```
% Centre of Mass coordinates
% In solidworks, click calculate and then click physical properties
% You should only modify the numeric values of this template (initially 0).
% The rest of text should remain unchanged.

X=0.00; % X-Coordinate x of the center of mass (mm, 2 decimal places) - 0.4 points
Y=0.00; % Y-Coordinate of the center of mass (mm, 2 decimal places) - 0.4 points
Z=0.00; % Z-Coordinate of the center of mass (mm, 2 decimal places) - 0.4 points
Mass=0.00; % Solid mass (in grams, 2 decimal places) - 4 points
Area=0.00; % Solid Surface Area (in mm^2, 2 decimal places) - 0.8 points

% The orientation of the center of mass will be evaluated with 4 points.
```

Figure 8. Doctus-Matlab template.

Doctus, by executing a Matlab code snippet, corrects the exercise by comparing the student submissions with the theoretical results from the table previously created with the physical properties (with an allowable geometric tolerance that has also been a subject of study).

The proposed methodology for validation geometric integrity—based on the comparison of mass properties, center of gravity, and surface area—demonstrates high computational robustness. The statistical probability of geometric false positive (distinct solids with identical physical properties) is estimated to be below 0.1%, a claim justified by the high dimensionality of the feature vector used (mass, area surface, X center of mass coordinate, Y center of mass coordinate, Z center of mass coordinate); assuming a standard floating-point precision of 10^{-2} for each descriptor, the cardinality of the feature space reaches 10^{10}, making the probability of accidental collisions stochastically negligible. By integrating zero-order, first-order, and surface descriptors [29], the system generates a unique shape fingerprint as a geometric hash [30,31]. In the context of engineering models constrained by functional requirements, the likelihood of two distinct models yielding identical descriptors within the system's tolerance is considered practically zero, thereby ensuring the pedagogical integrity and objectivity of the evaluation framework.

Doctus gives feedback to the students to make it easier for them to find their mistakes, including a grade and some tips (previously programmed by the professor, based on his knowledge and experience). An example is shown in Figure 9.

```
Values assessment. Results list:
Position of the centre of mass: (0/1)
Orientation X-Coordinate of CoM: (0/1)
Orientation Y-Coordinate of CoM (0/1)
Orientation Z-Coordinate of CoM (0/1)
Mass: (1/1)
Area: (0/1)
```

Figure 9. Sample feedback sent to the students.

Students can see the feedback from Doctus, and they decide whether to repeat it or not.

To bridge the gap between automated numerical validation and the pedagogical objectives described in Section 2, the system's metrics are mapped directly to specific technical competencies. While physical properties—such as mass, volume, and surface area—serve as the primary data points for grading, they function as proxies for the student's mastery of CAD modeling principles. Table 2 illustrates this traceability, detailing how each automated parameter validates a particular micro-competency and its underlying pedagogical justification.

Table 2. Traceability between parameters and competencies.

Automated Parameter	Evaluated Micro-Competency	Pedagogical Justification
Mass	Dimensional accuracy	Ensures that the 2D profile and 3D operations (extrude/revolve/swept/...) are executed with precision.
Center of Mass (X, Y, Z)	Spatial orientation & Constraints	Validates the student's understanding of the origin, reference planes, and geometric constraints.
Surface area	Detail processing	Detects errors in secondary features like filets, chamfers, or shells that may not significantly alter volume.
9/10 success rule	Autonomy & rigor	Promotes self-criticism and systematic verification before final submission.

2.4. Application of Automated Assessment in CAD Courses

At our university, automated assessment technology has been applied in CAD courses in the first years of engineering studies. For safety and to guarantee fair grading, manual revisions have been made to contrast the automatic correction. Some problems have become known during this time and have been analyzed and solved. The algorithm has been tested sufficiently to prove its reliability.

Automated assessment has reduced professors' time spent on correction, so they have more time to solve doubts and work closely with students, attending to them more personally.

The saved time has also been invested in creating new parameterized exercises, so they now have many 3D model voluntary tasks to practice and homework that is part of the course evaluation. At present, students can choose from a substantial repository of exercises and submit their CAD jobs on their own working schedule to prepare for the exam. They will have immediate feedback after each submission, so they can correct their errors and continue studying without interruption.

The correlation between Doctus results and success in passing the course has made students highlight the importance of working on continuously evaluated homework as a formative assessment, and it motivates them to study.

3. Results

Our university adopted this methodology 11 years ago, and since its implementation, approximately 500 students have utilized it to assess over 13,500 submissions across more than 200 individual exercises. However, to ensure data consistency and validity, the analytic dataset was strictly restricted to $N = 441$ independent observations from the last six academic cohorts. We systematically excluded "no-shows" (students lacking a final face-to-face exam grade) and records from early pilot years where critical continuous metrics (such as the optional submission rate) were not yet structurally logged.

The system incorporates automated grading and allows for resubmissions, facilitating a streamlined assessment process. Continuous development over the years has enhanced the tool at all levels, ensuring an intuitive experience. This continuous assessment process has enabled the collection of extensive data on student performance. To provide complete transparency regarding the model specification and the operationalization of the dataset, the recorded variables are formally defined in Table 3.

Table 3. Data dictionary and variable operationalization.

Variable	Definition	Role in Model	Original Scale	Scale in Model
exam_grade	Final face-to-face exam grade.	Dependent Variable	Continuous [0, 10]	Continuous [0, 1]
doctus_grade	Cumulative score obtained on the Doctus platform.	Predictor	Continuous [0, 10]	Continuous [0, 1]
submission_rate	Ratio of optional exercises successfully completed.	Predictor	Percentage [0%, 100%]	Continuous [0, 1]
expectations	Continuous index reflecting the evolving methodological rigor and complexity required by instructors.	Predictor	Percentage [0%, 100%]	Continuous [0, 1]
phase	Phasing methodology.	Predictor (Excluded after VIF analysis)	Binary (0/1)	Binary (0/1)
academic_cohort	Set of six mutually exclusive dummy variables (from '18–19' to '23–24') indicating the student's enrollment year.	Predictor (Excluded after VIF analysis. Used as Clustering Variable)	6 Binary Dummies (0/1)	Ordinal [1, 6]

https://doi.org/10.3390/app16104866

It is important to clarify the exact handling of the temporal dimension (the student's academic cohort). Natively, this information was extracted from the platform as a set of six mutually exclusive dummy variables. To accurately model the hierarchical structure of the data, these binary indicators were first mathematically collapsed into a single ordinal sequence (ranging from 1 to 6). This ordinal variable was subsequently normalized to a [0, 1] scale strictly for the initial collinearity (VIF) diagnostics. However, for the final econometric estimation, the unnormalized ordinal sequence was utilized to define the exact discrete groups required for computing the cluster-robust standard errors.

All continuous features were mapped to a [0, 1] scale via Min-Max scaling. It is important to note that the original variables already possessed natural and well-defined bounds (e.g., submission rates bounded between 0% and 100%, and academic grades bounded between 0 and 10). Consequently, mapping them to a [0, 1] interval preserves their intuitive interpretability as fractional achievements. This linear transformation fulfills two critical roles: firstly, it allows for direct evaluation and comparability of the Ordinary Least Squares (OLS) coefficients; secondly, it ensures the numerical stability of the non-linear machine learning models deployed in subsequent phases of the study.

All statistical analyses, econometric estimations, and machine learning models were implemented in Python 3.12. Specifically, linear and logistic regressions, along with all post-estimation diagnostics, were computed using statsmodels [32], a highly consolidated framework for rigorous econometric modeling. Conversely, non-linear modeling, hyperparameter optimization, and cross-validation procedures were executed via scikit-learn [33], the industry standard for predictive data analysis.

Several analyses were conducted to understand the relationship between continuous assessment variables and the final exam grade. To ensure model stability and interpretability, a diagnostic of multicollinearity was performed using Variance Inflation Factors (VIF). In the initial model specification, which included all potential predictors, severe multicollinearity was detected. As shown in Table 4, variables such as 'expectations', phase', and 'academic cohort' exhibited VIF values significantly above the standard threshold of 10. This statistical overlap is expected, as these variables concurrently capture the same underlying latent construct: the evolution and maturation of the course's teaching methodology and the refinement of the assessment tool over time.

Table 4. VIF factors in the original model.

Feature	VIF Factor
doctus_grade	7.80
submission_rate	15.03
Expectations	25.30
Phase	13.94
academic_cohort (normalized)	11.75

To resolve this issue and achieve a parsimonious model, a strategic variable removal was implemented. 'expectations' was retained as the primary proxy for methodological maturity, while the redundant academic cohort and phase indicators were excluded. The final VIF analysis resulted in a significantly more stable configuration ('doctus_grade' = 7.68; 'submission_rate' = 13.55; 'expectations' = 5.18). Although 'submission rate' maintains a VIF slightly above 10, the coefficients remain interpretable and statistically significant.

Following the resolution of multicollinearity, the definitive multilinear regression model was estimated. Recognizing the hierarchical structure of the dataset (where students are nested within distinct academic cohorts), the model was specified using Cluster-Robust

Standard Errors [34] grouped by academic year. This methodological choice mathematically corrects for intra-group correlation and ensures conservative, reliable significance estimates. The results of this robust specification are presented in Table 5.

Table 5. Final multilinear regression model.

Variable	Coef. (β)	Robust Std. Err.	*z*-Statistics	*p*-Value	95% Conf. Interval
(Constant)	−0.449	0.110	−4.07	<0.001	[−0.665, −0.233]
doctus_grade	0.242	0.067	3.61	<0.001	[0.111, 0.374]
submission_rate	0.338	0.084	4.00	<0.001	[0.173, 0.504]
expectations	0.729	0.098	7.43	<0.001	[0.536, 0.921]

Notes: Dependent variable: Exam Grade. $N = 441$. Model Fit: $R^2 = 0.369$ (Adjusted $R^2 = 0.365$). Overall Model Significance: F-statistic = 115.0 ($p < 0.001$). Standard errors are robust to intra-group correlation (clustered by academic cohort). Residual diagnostics: Jarque–Bera $p = 0.098$.

As presented in Table 5, the final robust model explains 36.9% of the variance in the target variable ($R^2 = 0.369$) and demonstrates high overall significance (*F*-statistic = 115.0, $p < 0.001$). Despite the more rigorous standard error estimates imposed by the cluster correction, all independent variables remain highly statistically significant ($p < 0.001$). The resulting model is represented by the following equation:

$$exam\_grade = -0.449 + 0.242 \cdot doctus\_grade + 0.338 \cdot submission\_rate + 0.729 \cdot expectations$$

It is important to reiterate that all predictive variables in this equation are scaled to a [0, 1] interval. Consequently, these coefficients represent the absolute maximum impact (a transition from 0% to 100% achievement) of each metric on the final exam score. The magnitude of the coefficients reveals critical pedagogical insights. The maturity of the teaching methodology (‘expectations’) exerts the strongest positive association with final exam performance (β = 0.729). Furthermore, student engagement metrics independently predict success: maintaining a high rate of optional submissions (β = 0.338) and achieving a solid continuous assessment score (β = 0.242) significantly boosts the final outcome. The 95% confidence intervals confirm that these positive effects are stable and strictly bounded away from zero.

Finally, a diagnostic analysis was conducted to verify compliance with the underlying assumptions of the OLS model. The Shapiro–Wilk test indicated no strong evidence to reject the normality of the residuals ($p = 0.14$), a conclusion further corroborated by the Jarque–Bera statistic ($p \approx 0.10$). Linearity was visually confirmed via fitted-versus-residuals plots. While the Breusch–Pagan test indicated the presence of heteroscedasticity (*p*-value = 0.032), this finding ultimately validates our proactive methodological decision to employ cluster-robust standard errors. This approach asymptotically neutralizes both heteroscedastic and intra-cluster correlation biases, ensuring the validity of the standard errors and the reliability of the reported inferences [34].

To further evaluate the pedagogical impact in binary terms (success versus failure), a Logistic Regression (Logit) model was estimated to predict the probability of a student passing the final exam (where ‘exam_grade’ $\geq$ 0.5 in the normalized scale, equivalent to $\geq$5.0 out of 10). The model parameters, estimated via Maximum Likelihood (MLE), are presented in Table 6. The overall model is highly significant (Likelihood Ratio test $p < 0.001$), demonstrating that continuous assessment metrics are robust predictors of academic survival.

Table 6. Logistic regression model.

Variable	Coef. (β)	Std. Err.	*z*-Statistics	*p*-Value	Odds Ratio (e^{β})	95% Conf. Interval
(Constant)	−6.119	0.752	−8.13	<0.001	0.002	[−7.593, −4.645]
doctus_grade	1.597	0.535	2.99	0.003	4.936	[0.549, 2.644]
submission_rate	2.007	0.605	3.32	0.001	7.439	[0.822, 3.192]
expectations	4.809	0.769	6.26	<0.001	122.56	[3.302, 6.315]

Notes: Dependent variable: Exam Pass (1 = Pass, 0 = Fail). $N = 441$ (204 Pass, 237 Fail). Model Fit: McFadden's Pseudo $R^2 = 0.186$; Log-Likelihood = −247.80. Overall model significance: LLR p-value < 0.001.

The translation of the log-odds coefficients into Odds Ratios (e^{β}) provides critical, actionable pedagogical insights. Because the predictors are mapped onto a [0, 1] scale, the reported Odds Ratios reflect the theoretical multiplicative effect of transitioning from the absolute minimum (0) to the absolute maximum (1) of a given metric.

For instance, maximizing the optional exercise flow ('submission_rate' transitioning from 0% to 100%) multiplies a student's odds of passing the exam by a factor of 7.4 ($p = 0.001$). Similarly, achieving a perfect continuous assessment score ('doctus_grade') increases the odds of passing by nearly a factor of 5 ($p = 0.003$). Finally, the high odds ratio for 'expectations' mathematically captures a macro-level cohort effect: the platform's utility as a catalyst for passing the exam is amplified in academic years subjected to the highest levels of methodological rigor.

Subsequently, to explore potential non-linear relationships and interactions, a Decision Tree Regressor (CART) was trained [35]. Hyperparameters were optimized via Grid Search with 5-fold cross-validation. The search space evaluated mathematically valid ranges, including maximum depth [3, 5, 7, 10, None], minimum samples per leaf [1, 2, 4], and minimum samples to split a node [2, 5, 10]. The optimization process yielded a final hyperparameter configuration comprising a maximum depth of 3, a minimum of 1 sample per leaf, and a minimum of 2 samples required to split an internal node.

To ensure absolute reproducibility and structural stability against training-test split variance, a 100-iteration Monte Carlo cross-validation was implemented utilizing the previously optimized hyperparameters. The mean performance across 100 random seeds ($R^2 = 0.318 \pm 0.093$, with a 95% confidence interval of [0.136, 0.500]) demonstrated that the model's predictive capacity is stable. This relatively wide confidence interval is an expected mathematical reflection of the inherent heterogeneity among distinct student cohorts and the multifactorial nature of academic performance across different academic years.

The optimized non-linear model achieved a maximum explanatory power similar to the robust linear model. This confirms that the remaining unexplained variance is not an artifact of a rigid linear specification but rather reflects unobservable latent human factors inherent to educational settings. Variables such as innate spatial intelligence, prior technical background, study time outside the platform, and exam-day anxiety naturally limit the theoretical maximum R^2. The authors posit that, within the complex context of engineering education, explaining over a third of the variance solely through automated platform engagement constitutes a highly robust and actionable result. This perspective is supported by foundational methodological standards in educational and behavioral sciences, which establish that an explained variance exceeding 26% represents a large and practically significant effect size [36], particularly when considering the multitude of unobservable human factors that inherently influence academic performance.

Beyond predictive accuracy, the primary value of the CART algorithm lies in its high interpretability. Having successfully validated the model's generalization capacity via Monte Carlo cross-validation, the objective shifts from performance estimation to the extraction of actionable pedagogical rules. To capture the most precise decision thresholds, the definitive model was trained on the complete analytic dataset ($N = 441$). Each node in the diagram displays the splitting criterion, the number of students meeting that condition ('samples'), and the predicted normalized exam grade ('value'). Following the branches downward based on the true/false conditions leads to the final predicted outcome. For instance, tracing the path of a highly engaged student involves navigating the right branch of the root node ('submission_rate' > 0.64); if this student also belongs to a cohort with high methodological rigor ('expectations' > 0.85) and maintains a solid continuous assessment ('doctus_grade' > 0.55), the path terminates at the far-right leaf, predicting an outstanding normalized exam grade of 0.90. This definitive hierarchical structure is presented in Figure 10.

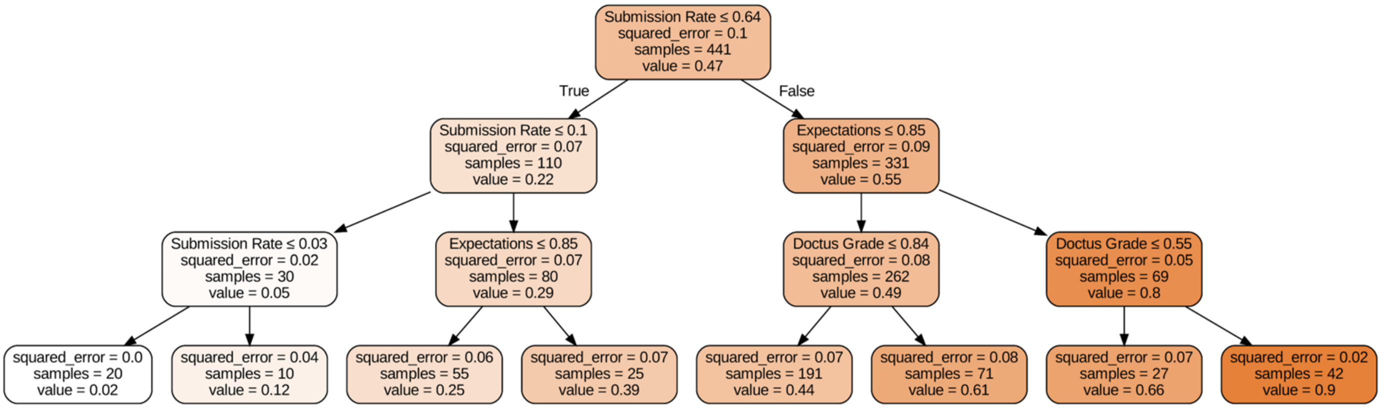

Figure 10. Definitive pedagogical decision tree (CART) for predicting normalized final exam performance.

The root node identifies the submission rate as the primary splitting criterion, creating two main subsets. However, the authors wish to emphatically stress that this relationship is purely observational. While highly predictive, these decision boundaries must be strictly interpreted as robust correlational associations rather than direct causal mechanisms. The submission rate likely functions as an intermediate indicator of student engagement and self-selection; highly motivated students consistently complete optional tasks, which in turn strongly predict higher exam scores.

By traversing the decision paths, several distinct student profiles emerge, transforming the tree into actionable pedagogical rules:

- Disconnected profile: Students completing less than 10% of optional assignments are immediately isolated by the model, predicting a severe failure (Max. exam grade ≈ 1.2 out of 10), regardless of other variables.
- Inconsistent profile: Students with a submission rate below 64% generally fail to internalize the concepts, resulting in a predicted failure (Max. exam grade ≈ 3.9).
- Committed profile: Surpassing the 64% submission threshold is the critical engagement indicator. Under normal expectations, these students secure passing grades if their continuous platform performance is adequate (Exam grade ≈ 6).
- High-performance profile: Notably, the model reveals a positive interaction between course rigor and student effort. In cohorts with high instructor expectations, students who maintain a submission rate above 64% achieve the highest predicted outcomes if their continuous platform performance is adequate (Exam grade ≈ 9).

To further explore potential predictive limits, a Random Forest regressor [37] was employed. An extensive grid search (CV = 5) optimized hyperparameters (200 estimators, max depth of 3, min samples leaf of 6). The evaluation of this ensemble model yielded an

R^2 of 0.384 on the test set. Since the ensemble did not significantly outperform the single decision tree but completely sacrificed interpretability, it was discarded in favor of the CART model.

A fundamental limitation of these observational models, which naturally constrain the maximum explanatory power, is the absence of variables measuring innate spatial aptitude or prior foundational knowledge. While the Doctus platform's methodology is positively related to academic success, the final exam grade inherently depends on unobserved student qualities and self-regulation skills outside the platform's scope. Furthermore, a critical avenue for future research is external validation; applying this predictive architecture to distinct engineering institutions will be vital to ascertain the cross-contextual generalizability of these findings.

4. Discussion

Doctus is a tool that has been operating consistently since 2007, and now, in this work, it has been adapted for overcoming new 3D models evaluation challenges with no limits on complexity. To overcome these challenges, it has been necessary to adapt the Doctus engine itself and to establish the right values for ensuring 3D models are feasible, equivalent and sufficiently variable, while avoiding false negatives and positives. This tool has been running in the last 8 years in the CAD subject in our university where more than 13.500 tasks have been evaluated. Based on this experience, we conclude that the Doctus-CAD environment has become a robust and reliable tool for addressing complex systems and is appropriate for university-level studies.

A significant advantage of the proposed methodology is its interoperability regarding student workflows. While the generation of exercises and the final evaluation core are centralized within the Doctus platform using the SolidWorks API as the golden standard for geometric validation, the student's modeling process remains entirely software-agnostic. Provided that students apply a standard density of 1000 kg/m^3 to their models, the system can accurately verify geometries authored in diverse environments such as OnShape, Autodesk Fusion 360, Solid Edge (Siemens), Catia (Dassault Systemes), Blender, or, among others, Rhino 3D.

This distinction is critical: the platform offers students the pedagogical freedom to utilize modern cloud-based or hybrid tools while maintaining a rigorous and unified grading benchmark through its backend integration. Thus, the methodology combines the robustness of industry-standard APIs with the flexibility required by current diverse educational CAD ecosystems.

Some data analysis methods have been conducted to evaluate the effect of using our approach on the final exam scores of students. Both the multiple linear regression model and the decision tree model can explain approximately 38% of the variance in the data, considering the ratio of optional submissions completed, the instructors' expectations, and the grade obtained in Doctus. While this is not a definitive indicator, the analysis helps instructors identify patterns in student progress, allowing them to anticipate potential difficulties and failures at the end of the course.

Further work is expected in several areas. First, a deeper exploration of machine learning models is necessary, which would involve incorporating additional data to improve the early detection of potential failure cases [38]. Specifically, adding demographic variables and academic background indicators, such as entrance grades or performance in other technical subjects, could provide a more comprehensive view of the students' profiles. Second, a quantitative analysis of the error produced by SolidWorks when similar shapes are designed by different operations, or the number of operations that give us valid results, to establish their limits in the proposed tasks. Third, develop a large library of models "ready to be proposed" or "available to the student" for testing or improvement. And finally, improve the feedback given to the students to clarify where their mistakes are or could be when the grades are not as expected.

5. Conclusions

In this work, a new tool for automatic CAD (Computer-Aided Design) exercise evaluation has been presented. To achieve this aim, new algorithms have been developed in parallel, capable of generating and automatically grading personalized tasks for students through the interaction of SolidWorks 2025, VBA, Doctus, and MATLAB 2025b.

The aim of this work, rather than merely automating instructor tasks, is to become an important pedagogical aid for increasing student engagement, motivation, and success via active learning and instantaneous feedback. The instructor may focus on developing new learning paths for students, making them adaptive, challenging, and interesting. They may also free routine grading time in favor of more valuable face-to-face student interactions. Indeed, in times of pervasive automation, direct interpersonal support is the biggest differential factor in higher education.

It may also be interesting to relate the presented approach to the increasing use of interest in AI Large Language Models to support education and automated grading. While prompt-based LLM frameworks may be insufficient for the 3D-space processing discussed here, a look at the possibilities of introducing AI models in some parts of the workflows should likely result in valuable opportunities for improvement. This remains an open research field.

Author Contributions: Conceptualization, F.S.-M. and F.G.-E.; Methodology, F.S.-M. and M.A.; Software, F.S.-M. and F.G.-E.; Validation, F.S.-M., M.A., and F.G.-E.; Resources, F.V.-G.; Data curation, D.C.; Writing—original draft, F.S.-M., M.A., D.C., F.G.-E., and F.V.-G.; Writing—review & editing, F.S.-M., M.A., D.C., F.G.-E., and F.V.-G. All authors have read and agreed to the published version of the manuscript.

Funding: The APC were funded by Universidad Loyola Andalucía. The authors would also like to acknowledge the support provided by the University of Seville through its institutional discount program.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Due to the privacy of student academic records, the datasets generated and analyzed during the current study are not publicly available but are available from the corresponding author on reasonable request.

Acknowledgments: The authors would like to thank all the students who helped improve the tools through their valuable questions and comments. We also wish to acknowledge Carlos Sánchez for his work in adapting the Doctus platform to accommodate externally generated content, such as the SolidWorks-based task descriptions used in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Buckley, J.; Seery, N.; Canty, D. A Heuristic Framework of Spatial Ability: A Review and Synthesis of Spatial Factor Literature to Support its Translation into STEM Education. *Educ. Psychol. Rev.* **2018**, *30*, 947–972. [CrossRef]
2. Wai, J.; Lubinski, D.; Benbow, C.P. Spatial ability for STEM domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. *J. Educ. Psychol.* **2009**, *101*, 817–835. [CrossRef]
3. Gittinger, M.; Wiesche, D. Systematic review of spatial abilities and virtual reality: The role of interaction. *J. Eng. Educ.* **2024**, *113*, 919–938. [CrossRef]

4. Maeda, Y.; Yoon, S.Y. A Meta-Analysis on Gender Differences in Mental Rotation Ability Measured by the Purdue Spatial Visualization Tests: Visualization of Rotations (PSVT:R). *Educ. Psychol. Rev.* **2013**, *25*, 69–94. [CrossRef]
5. Roda-Sales, A.; Gracia-Ibáñez, V.; Agost, M.-J.; Gómez, M.G.-F.; Núñez-García, M. Introducing Parametric CAD in a First Year Course in Engineering Degree: A Case Study. In *Advances in Design Engineering II*; Springer: Cham, Switzerland, 2020; pp. 411–419. [CrossRef]
6. Duan, X.-R. The Required Knowledge and Skill Competencies for CAD and Design Engineering Students. *Chemistry* **2006**, *2*, 11.
7. de la Peña, D.M.; Gómez-Estern, F.; Dormido, S. A new Internet tool for automatic evaluation in Control Systems and Programming. *Comput. Educ.* **2012**, *59*, 535–550. [CrossRef]
8. Ferreira, R.; Canesche, M.; Jamieson, P.; Neto, O.P.V.; Nacif, J.A.M. Examples and tutorials on using Google Colab and Gradio to create online interactive student-learning modules. *Comput. Appl. Eng. Educ.* **2024**, *32*, e22729. [CrossRef]
9. Calderon, K.; Serrano, N.; Blanco, C.; Gutierrez, I. Automated and continuous assessment implementation in a programming course. *Comput. Appl. Eng. Educ.* **2024**, *32*, e22681. [CrossRef]
10. de la Peña, D.M.; Domínguez, M.; Gomez-Estern, F.; Reinoso, Ó.; Torres, F.; Dormido, S. Estado del arte de la educación en automática. *Rev. Iberoam. Automática Inform. Ind.* **2022**, *19*, 117–131. [CrossRef]
11. Ruesgas, L.G.; Gual, F.V.; Torrecillas, C.; Vázquez, A.V. Propuestas Tecnológicas de Autocorrección de ejercicios de modelado 3D = Technological Proposals for self-assessment of 3D modelling exercises. *Adv. Build. Educ.* **2020**, *4*, 42–59. [CrossRef]
12. Eltaief, A.; Amor, S.B.; Louhichi, B.; Alrasheedi, N.H.; Seibi, A. Automated Assessment Tool for 3D Computer-Aided Design Models. *Appl. Sci.* **2024**, *14*, 4578. [CrossRef]
13. Yoon, Y.; Jeon, Y.; Kim, J.; Han, S.; Kim, H.; Kwon, S. CADuBoost: Enhancing Education in Mechanical 3D CAD Modeling Through Automated Grading and Feedback System. *Comput. Appl. Eng. Educ.* **2025**, *33*, e70096. [CrossRef]
14. Jianwu, L.W.; Yew, L.S.; On, L.K.; Keong, T.C.; Sheng, R.T.Y.; Bin Sani, S.; Agnes, T.H.J. Artificial intelligence-enabled evaluating for computer-aided drawings (AMCAD). *Int. J. Mech. Eng. Educ.* **2024**, *52*, 3–31. [CrossRef]
15. Tiainen, T.; Jaakma, K.; Kiviluoma, P.; Viitala, R.; Kuosmanen, P. An automatic assessment system for CAD education. In *Proceedings of the SEFI 47th Annual Conference*; Nagy, B.V., Murphy, M., Jarvinen, H.-M., Kálmán, A., Eds.; Société Européenne Pour la Formation des Ingénieurs: Brussels, Belgium, 2020; pp. 1190–1200. Available online: https://www.sefi.be/wp-content/uploads/2019/10/SEFI2019_Proceedings.pdf#page=1190 (accessed on 6 May 2026).
16. Jaakma, K.; Kiviluoma, P. Auto-assessment tools for mechanical computer aided design education. *Heliyon* **2019**, *5*, e02622. [CrossRef]
17. Ericsson, K.A.; Krampe, R.T.; Tesch-Römer, C. The role of deliberate practice in the acquisition of expert performance. *Psychol. Rev.* **1993**, *100*, 363–406. [CrossRef]
18. Barton, C. *How I Wish I'd Taught Maths: Lessons Learned from Research: Conversations with Experts, and 12 Years of Mistakes*; Hachette UK Distribution: Oxfordshire, UK, 2018; ISBN 9781911382492.
19. Rico, M.A.; Moreno, J.J.G.; Salmerón-Medina, F. *Representación Gráfica de Sistemas Mecatrónicos*; Ediciones Paraninfo: Madrid, Spain, 2021.
20. Menary, G.H.; Robinson, T.T. Novel approaches for teaching and assessing CAD. In Proceedings of the International Conference on Engineering Education ICEE2011, Madinah, South Arabia, 25–27 December 2011; pp. 21–26. Available online: https://www.researchgate.net/profile/Gh-Menary/publication/228452005_Novel_approaches_for_teaching_and_assessing_CAD/links/09e4150af190ccb410000000/Novel-approaches-for-teaching-and-assessing-CAD.pdf (accessed on 6 May 2026).
21. Yue, J. Teaching techniques in computer aided design. In *Proceedings of the 1999 Annual Conference, Charlotte, NC, USA, 20–23 June 1999*; ASEE: Washington, DC, USA, 1999; pp. 4.494.1–4.494.6. Available online: https://peer.asee.org/7979.pdf (accessed on 6 May 2026).
22. Brown, G.A.; Bull, J.; Pendlebury, M. *Assessing Student Learning in Higher Education*; Routledge: London, UK, 2013.
23. Race, P. *A Practical Guide to Assessment, Learning and Teaching*; Routledge: New York, NY, USA, 2015.
24. Sancho-Vinuesa, T.; Escudero-Viladoms, N. A Proposal for Formative Assessment with Automatic Feedback on an Online Mathematics Subject. *RUSC Rev. Univ. Soc. Conoc.* **2012**, *9*. [CrossRef]
25. Aparicio-Gómez, O.-Y.; Ostos-Ortiz, O.-L. *Evaluación Formativa*; Universidad Santo Tomás: Bogotá, Colombia, 2020.
26. Brink, H.; Kilbrink, N.; Gericke, N. Teach to use CAD or through using CAD: An interview study with technology teachers. *Int. J. Technol. Des. Educ.* **2023**, *33*, 957–979. [CrossRef]
27. Wood, D.; Bruner, J.S.; Ross, G. The role of tutoring in problem solving. *J. Child Psychol. Psychiatry* **1976**, *17*, 89–100. [CrossRef]
28. Zhang, J. Scaffolding Probabilistic Reasoning in Civil Engineering Education: Integrating AI Tutoring with Simulation-Based Learning. *Educ. Sci.* **2026**, *16*, 103. [CrossRef]
29. Minkowski, H. Volumen und Oberfläche. In *Ausgewählte Arbeiten zur Zahlentheorie und zur Geometrie: Mit D. Hilberts Gedächtnisrede auf H. Minkowski, Göttingen 1909*; Springer: Berlin/Heidelberg, Germany, 1989; pp. 146–192.
30. Osada, R.; Funkhouser, T.; Chazelle, B.; Dobkin, D. Shape distributions. *ACM Trans. Graph. (TOG)* **2002**, *21*, 807–832. [CrossRef]

31. Zhang, D.; Lu, G. Review of shape representation and description techniques. *Pattern Recognit.* **2004**, *37*, 1–19. [CrossRef]
32. Seabold, S.; Perktold, J. Statsmodels: Econometric and statistical modeling with python. In Proceedings of the 9th Python in Science Conference, Austin, TX, USA, 28 June–3 July 2010; Volume 7, pp. 92–96.
33. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
34. Cameron, A.C.; Miller, D.L. A practitioner's guide to cluster-robust inference. *J. Hum. Resour.* **2015**, *50*, 317–372. [CrossRef]
35. Breiman, L.; Friedman, J.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2017.
36. Cohen, J. *Statistical Power Analysis for the Behavioral Sciences*; Routledge: New York, NY, USA, 2013.
37. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
38. Arashpour, M.; Golafshani, E.M.; Parthiban, R.; Lamborn, J.; Kashani, A.; Li, H.; Farzanehfar, P. Predicting individual learning performance using machine-learning hybridized with the teaching-learning-based optimization. *Comput. Appl. Eng. Educ.* **2023**, *31*, 83–99. [CrossRef]