Simulations for the Precise Modeling of Exercises Including Time, Grades and Number of Attempts

Alberto Jiménez-Macías*, Pedro J. Muñoz-Merino, Carlos Delgado Kloos

Universidad Carlos III de Madrid, Leganés (Spain)

* Corresponding author: albjimen@it.uc3m.es

Received 10 November 2022 | Accepted 24 May 2024 | Published 9 October 2024

unir LA UNIVERSIDAD EN INTERNET

Abstract

Students' interactions with exercises can reveal interesting features that can be used to redesign or effectively use the exercises during the learning process. The precise modeling of exercises includes how grades can evolve, depending on the number of attempts and time spent on the exercises. A missing aspect is how a precise relationship among grades, number of attempts, and time spent can be inferred from student interactions with exercises using machine learning methods, and how it differs depending on different factors. In this study, we analyzed the application of different machine-learning methods for modeling different scenarios by varying the probability of answering correctly, dataset sizes, and distributions. The results show that the model converged when the probability of random guessing was low. For exercises with an average of 2 attempts, the model converged to 200 interactions. However, increasing the number of interactions beyond 200 does not affect the accuracy of the model.

Keywords

Content Modeling, Learning Analytics, Simulated Students, Smart Content.

DOI: 10.9781/ijimai.2024.10.002

I. Introduction

The learning content utilized in teaching and learning is crucial because it is a valuable tool for enhancing students' understanding and influencing their cognitive and metacognitive capacities. However, its usefulness may be limited if learning content remains stagnant and cannot be expanded. Smart learning content (SLC) included advanced features, such as adaptive personalization, sophisticated feedback forms, user authentication, learner modeling, data aggregation, and learning analytics [1]. SLC can improve student engagement and success by providing personalized learning experiences that are adapted to individual needs and preferences [2].

Depending on the type of content, we can have different types of student interaction. One type of content is related to tests in which students can make different attempts to solve problems, for example, multiple responses or fill-in-the-blank exercises. Probabilistic approaches, such as the Item Response Theory (IRT), have been utilized to model educational tests. This method allows the estimation of item characteristics, such as difficulty, discrimination, and guessing, using student interactions [3]. Some authors have also used content modeling to estimate additional content parameters. For example, some models help infer the skill acquired by students after using educational materials [4].

In this work, we focus on exercises and understanding such as any activity task that should be solved by a student at some time, after many attempts, and in which the student can achieve a grade for each attempt. Examples of types of exercises include multiple choices, multiple responses, and a drag&drop, but also an open problem in which automatic evaluation is not possible, and a teacher should grade it by looking at different steps and a long text.

An exercise can be better characterized by establishing a clear relationship between the number of attempts, time spent, or grades. In the results of the systematic literature review [5], these three exercise characteristics were the next most frequently used in different studies after excluding the three used by the IRT: difficulty, discrimination, and guessing. For example, Moreno-Marcos et al. [6] used grade, time, and number of attempts to identify behavioral patterns, such as persistence, efficiency, and constancy, within an intelligent tutoring system. Also, Feng et al. [7] calculated indicators of the activities carried out by students, including the average number of attempts for each question, time spent on the activities, and number of finished activities, among others.

The term "learning curve" in the field of education pertains to the speed at which a student acquires a specific skill or set of skills. Learning curves can be used to track a learner's advancement by evaluating their performance over time, identifying areas of strength and weakness, and determining the most effective means of supporting learning [8]. In the plots of this theory, the 2D graphs relate performance (the grade obtained) to the learning effort (the number of attempts), and performance to the time employed.

Please cite this article as:
A. Jiménez-Macías, P. J. Muñoz-Merino, C. Delgado Kloos. Simulations for the Precise Modeling of Exercises Including Time, Grades and Number of Attempts, International Journal of Interactive Multimedia and Artificial Intelligence, vol. 9, no. 3, pp. 127-140, 2025, http://dx.doi.org/10.9781/ijimai.2024.10.002

Different studies have analyzed these three characteristics independently in models for educational exercises; however, we identified a gap when using these three characteristics simultaneously in a model and analyzed how they relate to each other. The redesign and use of exercises can be improved by understanding the relationship between these indicators. The focus of this study was to use machine learning techniques to infer the relationships among these parameters based on student interactions with exercises.

To evaluate these exercise models, we needed a significant number of interactions performed by students in those exercises. One posible solution is to use simulated students. Previous studies have used different simulated students [9], [10], [11] to recreate different student learning situations. The use of simulations in this work does not attempt to replicate the real student's behavior, but to test the models in different predefined scenarios of student behavior so that we can know, for example, the number of interactions necessary for different cases. To evaluate the accuracy of machine-learning algorithms, different metrics can be used, such as precision, recall, F1, Root Mean Square Error(RMSE) default without normalization, and Area Under the Curve(AUC) [12].

This study aimed to analyze the possibility of using machine learning methods to infer these exercise indicators using student simulations. We propose the following research questions:

- Is it possible to obtain a time-grade-attempt model in exercises that are sufficiently accurate using traditional machine learning algorithms?
- How does the accuracy of the models vary for different types of exercises?
- What is the minimum number of interactions required to stabilize an exercise model with acceptable accuracy?
- How do different forms of student behavior modify the results obtained in the previous research questions?

This paper extends our paper [13]. We analyzed a model design for educational exercises using grade, number of attempts, and time spent. We tested different machine learning algorithms using simulated data for each variable using normal distributions. The paper is structured as follows: Sections I and II of this paper (Introduction and Related Work) include ideas from [13] but extend it with new ideas and references as the research questions have been extended to analyze the effect of changing the probability of answering correctly, dataset sizes and different distributions. Subsection III-A presents an overview of the extended paper [13] and takes ideas, results, and analysis from [13] but has been rewritten to try to increase clarity; Subsection III-B includes a new analysis of the respective metrics evaluated, the same visualizations with another dataset, and a new analysis of model over-fitting, while Sections IV, V, and VI are new and analyze the behavior of this proposed model in different scenarios. Section IV presents the simulations for different types of questions, dataset sizes, and distributions, Section V shows the results obtained in the simulations, Section VI presents a discussion of the results obtained, and Section VII presents the conclusions and future work.

II. Related Work

Smart learning environments (SLEs) are learning environments capable of enhancing education by using adaptive technologies [14]. The content available to the learner and the knowledge acquired by the learner is part of this environment. Content should be constructed based on the learner's previous experience by identifying their needs and learning styles [15]. Content modeling is relevant for learning because it allows for the redesign and improvement of teachers' content, thus helping students' learning.

In content modeling, probabilistic models take advantage of content parameters to understand and represent the learning materials. These models can be evaluated through simulations or real scenarios to provide insights into their effectiveness and adaptability in intelligent learning environments. The following subsections indicate the application of probabilistic models, the importance of parameters, simulations employed, and critical consideration of the number of interactions in optimizing these models to improve educational outcomes.

A. Probabilistic Models

Among the studies in which probabilistic methods are used, the most frequently used algorithm is IRT [16] [17] for modeling items in tests. IRT can estimate exercise parameters such as difficulty, discrimination, and guessing, based on the interactions made by the students in the questionnaires. For example, IRT was used to provide individual learning paths for students, which can alleviate disorientation and cognitive overload in learners based on the difficulty of course materials and their ability to improve learning efficiency and effectiveness [18]. The authors suggest that additional research and testing is required to thoroughly assess its effectiveness and potential limitations. Abbakumov [19] used a modified version of IRT to estimate the difficulty levels of items and address the cold-start problem using an application developed at the Higher School of Economics University. Consequently, learner motivation can be maintained, frustration and stress can be reduced, and learning outcomes can be improved. The author did not indicate any limitations but promised further work to evaluate the efficacy of the proposed model using real student data and to optimize the model's performance on topics with a medium level of difficulty, which typically has regression coefficients that are relatively inconsequential.

Artificial intelligence algorithms, such as regression, random forest, and neural networks, have been used to determine the parameters of Item Response Theory (IRT) and to evaluate the accuracy of these models [20] [21] [22]. In a study [23], a regression algorithm was used to measure difficulty and discrimination in multiple-choice questions. The results were compared to those obtained using IRT to estimate the same parameters. In addition, Lehman et al. [24] analyzed the emotions that students experience during conversation-based evaluations.

Another type of content modeling has been applied to discussion forums. Capuano et al. [25] used neural networks to classify students' answers in the forums and to detect the confusion perceived by students when participating in the discussion in real-time. The suggested approach can potentially enhance interactivity and support for students in Massive Open Online Courses (MOOCs). However, the authors acknowledge that additional research is necessary to assess the effectiveness of this approach thoroughly. Neural networks were used in discussion forums to detect feelings produced by a forum for students in MOOCs [26]. The linguistic-feature-based confusion classifier performed well on the evaluated metric F1-score, allowing real-time detection of message confusion. A limitation of the study was false negatives because teachers would not be able to identify messages in need of urgent intervention.

B. Parameters

In exercises, the grade, time spent, and the number of attempts have been used in different studies as indicators, as in the work by Feng, Heffernan and Koedinger [27]. Verdú et al. [28] proposed a model based on genetic algorithms and fuzzy systems to accurately classify questions according to their difficulty level in an intelligent tutoring system. They used the following parameters in their model: time in minutes from the last reading of the question to the delivery of the answer, grade obtained for that answer, and number of accesses or readings before sending the answer.

Regarding grade, Uto [29] proposed a model to estimate the ability and grade obtained by students in written essays using the IRT model with evaluator parameters integrated into a model of the topicality of the answers. This model is based on the Latent Dirichlet Allocation(LDA) model of responses obtained by students in essays. In addition, the final grades for a subject and master's degree in a university online mode were determined using different machine learning algorithms such as Naive Bayes, Decision Tree, Random Forest, and Neural Networks [30].

In terms of time, Rushkin, Chuang, and Tingle [4] described a log-normal model to estimate the slowness of the learners and the characteristics of the evaluations, such as discrimination and time intensity, using response times in an online course. In addition, Xue, Yaneva, Runyon and Baldw [20] predicted difficulty and response time for multiple-choice questions using information from each item text in the medical examination questions.

We [13] proposed providing more details on exercises based on three characteristics: grade of each attempt, time spent in each attempt, and number of attempts using the probabilistic method. The contribution of this study is to propose a detailed analysis of how the three indicators are related using simulations. In addition, they proposed different characteristics of the most used exercises, such as difficulty, discrimination, and guessing, calculated as parameters using IRT. The results demonstrated the accuracy of the machine learning algorithms using the proposed model design, indicating that the use of simulated students was a limitation of the study.

C. Simulations

Regarding simulation in education, VanLehn et al. [31] found three main applications in which simulated students could be used: as peers of real students, in instructional pedagogical design, and teachers' learning methods. Various tools have been developed, such as SimStudent [32] and Demonstr8 [33], to test different models in a simulated environment before testing them in a real environment. These tools are helpful for the learning process because they allow the evaluation of different conditions required in the evaluated models [34].

Moreover, some systems simulate the students during the learning process. For example, Graesser [35] proposed an architecture that uses a simulation approach to implement pedagogical agents that focus on peer learning. Vizcaino [36] described an architecture in a collaborative environment that uses simulated students to detect and avoid possible scenarios that do not improve collaborative learning.

We propose the use of student simulations to recreate different possible scenarios in which the model could be used. Previous studies used simulated students to validate the models proposed by the authors. For example, Champaign and Cohen [11] proposed an approach for selecting content in an intelligent tutoring system based on student interactions. A simulated student was used to validate the proposed model and attempt to recreate a real-world scenario. However, there are clear constraints in creating simulated students that exactly match real learners. The researchers determined that their algorithm was efficient in choosing relevant educational content for students by considering the prior learning experiences of similar peers. Dorcca [9] used simulated students to evaluate three strategies in models of student learning styles, reducing the number of resources needed to validate the proposed approaches, understanding the proposed system's behavior in this scenario, and making necessary changes to improve the design. However, simulated students may not fully capture real students' behavior and responses, and the effectiveness of adaptive educational systems with simulated students may not always be generalizable to real-world settings.

D. Number of Interactions

The number of interactions or runs required in any machine-learning algorithm is important to identify the performance of any proposed model and different studies have been conducted to identify the number of interactions or runs needed. For example, Liu et al. [37] found that it was necessary to run a Bayesian Network algorithm twice. Erickson et al. [38] identified 100 interactions to determine the best approach to learning object allocation. Frost and McCalle [39] required 25 simulations to determine the best performance among groups of learners. Riedesel et al. [40] performed 100 runs of simulations within an application to memorize basic techniques for students. BEETLE II [41] is a simulation-based physics tutor used to foster effective self-explanation in students, requiring 1000 simulation runs to find the best performance using the F-score metric.

In this context, this study contributes to the understanding of the minimum conditions necessary to test the proposed exercise model using simulations and to recreate the possible conditions in a real scenario. In addition, we provide information on the algorithms and minimum exercise interactions needed in the content model design so that other researchers can use these findings in other content such as discussion forums, archives, and wikis, used in any educational system.

III. Base Model for the Characterization of Exercises

Our previous work [13] proposed an exercise model based on interactions performed by students using machine learning algorithms. The model design was named the base model. We selected three characteristics mentioned in related works because the authors used them to characterize an exercise. Although these variables were used earlier, we aimed to understand better the relationship between grade, number of attempts, and time based on previous data on user interactions. The time and grade were based on the student's performance in each attempt. Fig. 1 shows the three characteristics using scatter plots to represent the grade and time for different attempts. The possible values for the three variables are graded with values between 0 and 10, time with values between 0 and n (representing the maximum possible value), and the number of attempts between 1 and m (representing the maximum possible value).

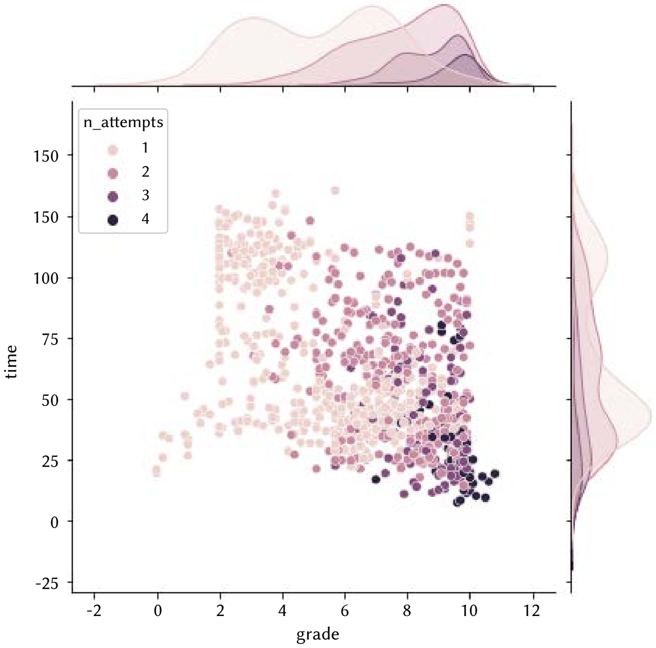
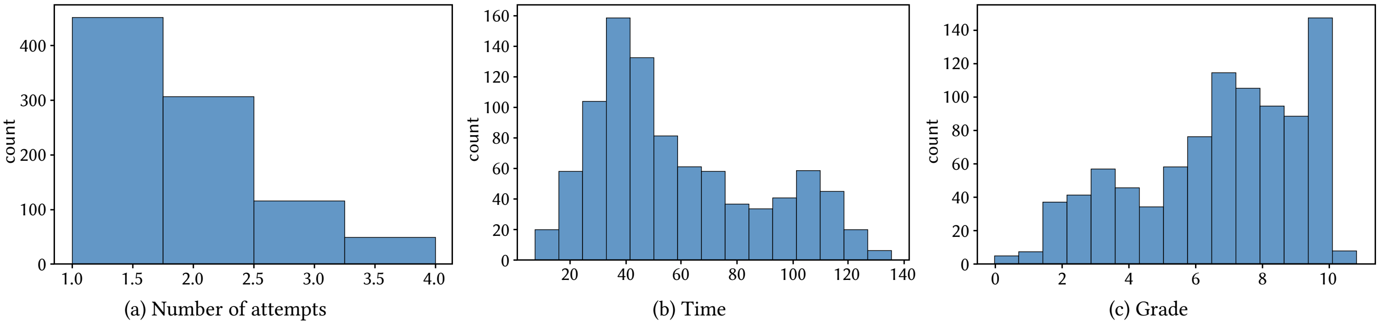


Fig. 1. Characterization of exercises.



(a) Number of attempts (b) Time (c) Grade

Fig. 2. Frequency distribution.

In this study, we used students' simulations to demonstrate their interactions and then trained different intelligence algorithms in the base model and three scenarios by modifying the probability of answering correctly, size of the data set, and distributions.

A. Simulation Using Base Model

We generated a dataset using a normal distribution with specific mean and standard deviation values for each variable including the number of attempts, grade, and time. We consider different criteria when generating each variable and their relationships. For instance, student's grades on the first attempt followed a normal distribution. For subsequent attempts, the same distribution was used but with a limit between the grade of the previous attempt and the maximum possible grade. However, no further attempts were made if students achieved their maximum grades. We used Python programming language with stats and random libraries.

Next, we conducted simulations based on three levels of student knowledge: easy, medium, and difficult. For each level, we adjusted the means and standard deviations of the variables (grade, time, and number of attempts) by increasing or decreasing their values in the distribution function depending on the level of previous knowledge.

We performed at least 150 simulation runs for an exercise with a probability of answering by guessing set at 0% and formed three groups of students categorized as low, medium, and high based on their previous knowledge. Each group comprised 150 students. The mean of the distributions shifted to the left or right depending on the student group. Each student group had a minimum of 150 interactions with the exercise on at least one attempt, and the students were allowed to perform multiple attempts. Simulations were used to train the model and determine the best curve representing the exercise characteristics.

The simulations aimed to recreate possible fictitious cases of student interactions but not to replicate real student behavior. Fig. 2 shows the frequency distribution of the generated variables, which are explained as follows:

- *Number of attempts*: Each student was assumed to have attempted at least one exercise. To preserve the randomness of the data, a random variable was calculated to establish the number of additional attempts that each student performed for that exercise. Subsequently, we performed validations for the second attempt, in which the obtained grade was randomized using a normal distribution, with the minimum value being the grade achieved in the previous attempt. Similarly, for the time variable, we set the randomness using a normal distribution, considering the maximum time obtained in the previous attempt, and ensured that the time did not exceed that of the previous attempt. We followed the same logic for subsequent attempts, such as the third, fourth, and beyond.
- *Grade*: We defined the students' grades obtained during the simulation from 0 to 10. For each attempt, we established a normal distribution, with the mean and standard deviation determined based on the three groups of students during the exercise. All the students had at least one grade for each exercise, as they had attempted it at least once. If students performed multiple attempts at exercise, each grade was obtained using a normal distribution between the maximum possible value for the exercise and the grade obtained on the previous attempt. However, if a student achieved their maximum grade, they were not allowed to make another attempt during the exercise. In all other situations, the new grade depended on students' number of attempts.
- *Time*: The exercise time of the trainees was limited from 0 to m seconds. The specific value of m depends on the difficulty level of the exercise, and in this study, we examined multiple values of m. To ensure that the data remained random, we created a normal distribution with mean and standard deviation values based on the three levels of exercise difficulty. As the number of attempts increases, the time variable decreases. However, as the grade level increased, the time variable also increased. If a student performs multiple exercise attempts, the time obtained is calculated randomly. This value was set as the maximum time calculated in a previous study.

B. Results Using Base Model

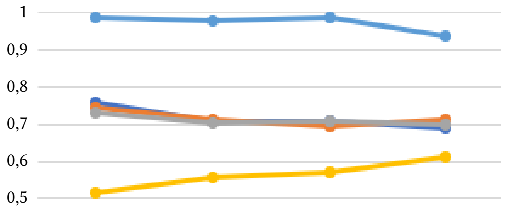
1. Curve Estimation Using Machine Learning

The base model was implemented using machine learning algorithms in Jupyter, using Python v3.9.2 as the programming language. The scikit library, an open-source library that implements many machine learning algorithms, was used. We used the same input dataset for all the algorithms and tested different classifier algorithms using 80% of the data for training and 20% of the simulated data for testing. The classifiers tested included Random Forest (with different depths), Logistic Regression, Nearest Neighbors (with different numbers of neighbors), Gaussian Naive Bayes, and Decision Tree (with different depths). We used grade as the dependent variable and the student's time spent on the exercises and the number of attempts as independent variables. To avoid overfitting, we used cross-validation with an algorithm to obtain the best metrics.

Fig. 3 shows the three best algorithms using precision(macro), recall(macro), f1(macro), RMSE, and AUC as metrics because the data were not balanced. The Nearest Neighbors with the $k = 10$ algorithm obtained relatively good metric values for approximating the relationship between the three variables used in the model design.

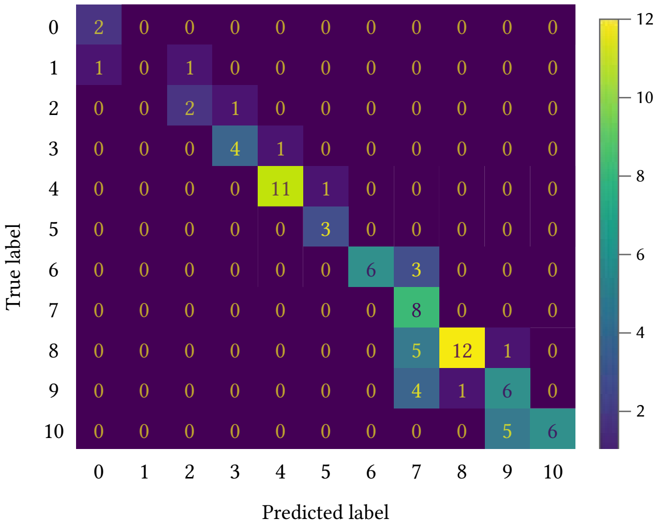
C. Best Algorithm Nearest Neighbors

We selected the best algorithm obtained in the previous section(i.e. the Nearest Neighbors with $k = 10$) and the *confusion_matrix* method of the sklearn.metrics library to obtain the confusion matrix. Fig. 4 shows the confusion matrix for the nearest neighbors algorithm with



		Nearst Neighbours (k = 10)	Decision Tree (Max Depth = 10)	Random Forest (Max Depth = 10)	MLP (tanh)
●	Precision (macro)	0,758	0,706	0,710	0,692
●	Recall (macro)	0,746	0,713	0,695	0,714
●	F1 (macro)	0,732	0,702	0,709	0,697
●	RMSE (macro)	0,515	0,558	0,571	0,610
●	AUC	0,99	0,980	0,990	0,940

Fig. 3. Metrics of algorithm model.



	0	1	2	3	4	5	6	7	8	9	10
0	2	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0
2	0	0	2	1	0	0	0	0	0	0	0
3	0	0	0	4	1	0	0	0	0	0	0
4	0	0	0	0	11	1	0	0	0	0	0
5	0	0	0	0	0	3	0	0	0	0	0
6	0	0	0	0	0	0	6	3	0	0	0
7	0	0	0	0	0	0	0	8	0	0	0
8	0	0	0	0	0	0	0	5	12	1	0
9	0	0	0	0	0	0	0	4	1	6	0
10	0	0	0	0	0	0	0	0	0	5	6

Fig. 4. Confunsion matrix with nearest neighbors(k=10).

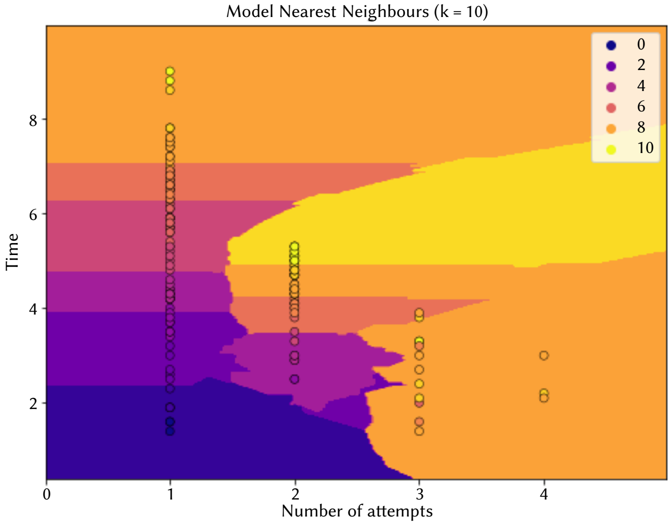


Fig. 5. Plot with nearest neighbors(k=10).

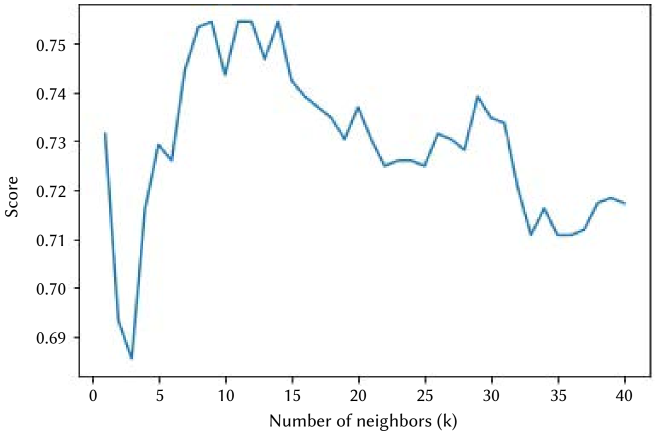


Fig. 6. Cross validation with different numbers of neighbors.

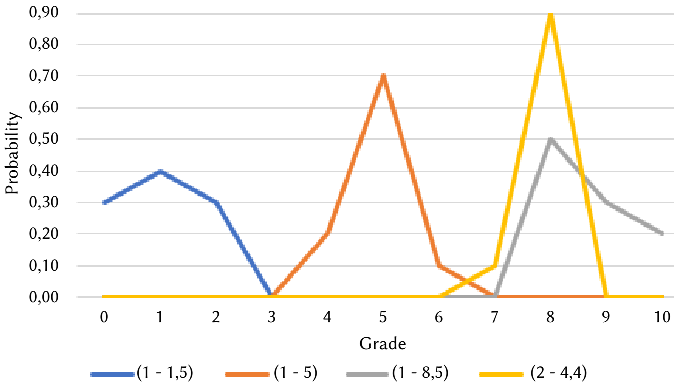


Fig. 7. Predicted probability with nearest neighbors(k=10).

k equal to 10. The matrix shows the different classes for the dependent variable grade of exercise. The figure shows the prediction accuracy for all grades, obtaining the highest values for grades 4 and 6. Also, the lowest accuracy was obtained for grades between 0 and 2.

The different clusters obtained in the model are shown in Fig. 5 for each attempt at different times. The changes between the colors indicate the Bayes decision boundary of the different classes corresponding to the dependent variable grade of colors ranging from blue for 0 to yellow for 10.

We used the *cross_val_score* method from the sklearn.model_selection library to prevent the overfitting of the best algorithm. Fig. 6 shows the results of cross-validation accuracy with five folds evaluated in the nearest neighbor algorithm using different neighbor values. The results indicate that the best values were obtained with neighbor values between 10 and 15, which helped avoid over-fitting and under-fitting. Therefore, the nearest neighbor algorithm with k equal to 10, which obtained the best-evaluated metrics is within this range.

In addition, we used the *predict_proba* method of the KNeighborsClassifier class within the sklearn library to predict the probability of all classes using grade as an independent variable. Fig. 7 shows the results using four elements of the test data set; the X-axis shows the different classes of the variable grade, whereas the Y-axis shows the probability estimate obtained in the predict_proba method. The test input data used were a pair of variables: the first corresponded to the number of attempts, and the second was the time spent.

The results show four examples tested in the selected algorithm in the first scenario (1–1.5): 1 is the number of attempts, and 1.5 is the time spent normalized between 0 and 10, obtaining a cumulative probability of 0.90 for a grade between 0 and 2. Finally, for a second attempt with a time of 4.4, a probability of 0.9 is obtained for the highest grade of 8. In summary, this exercise increased the time spent on the first attempt and the probability of improving grade.

The following sections aim to create different possible scenarios and analyze the performance of the machine learning algorithms to estimate the model using these characteristics. We tested different datasets with a different number of interactions, changing the probability of correctly answering the questions, and changing the distributions, and we used a group of students with the same prior knowledge.

IV. Simulations

In this section, we describe the methodologies used to generate different sets of simulated data for the three simulated datasets for each of the three exercise characteristics.

A. Using Different Probabilities of Answering Correctly

To illustrate the different probabilities of answering a question correctly, we used three types of questions used in student evaluations: 50% to represent true/false questions, 20% and 14% to represent multiple-choice questions with five or seven options and just one correct answer, and 7%and 5% to represent multiple-choice questions with six or seven options and two correct answers. The simulations aimed to understand the model's behavior for different types of questions depending on the probability of answering correctly, identifying changes in the model's accuracy, and whether more interactions are needed.

Using the data simulation, we used the same methods and libraries described in the previous section for the exercise model. We then ran 150 simulations corresponding to the interactions of 150 students during the exercise. Each student completed at least one interaction during the exercise and made more attempts in the same exercise. The grade in each of the simulated exercises was different because it depended on the type of probability.

- 50%: This type of probability corresponds to true/false questions. Students with no prior knowledge had a 50%probability of correctly answering
- 20% and 14%: These two probabilities represent questions with n options, of which only one was the correct answer. We simulated two random questions with a probability of a student answering randomly: 20% (one correct answer out of five options) and 14% (one correct answer out of seven options).
- 7% and 5%: In this type of probability, students had more possible selections because the exercise had a combination of n among m, where n is the number of correct answers and m is the number of choices. For the simulation, we considered two correct answers among the six options; we obtained a combination of 15 possibilities available to the student. Therefore, the probability of correctly answering the questions was 7%(1 out of 15). Finally, the other probability of 5%corresponds to a question with two correct answers among the seven options (1 out of 21).

If the student obtains the maximum grade on the first attempt or N attempts, the student makes no further attempts. The same conditions were used for the simulations in the base model.

B. Using Different Number of Interactions

Initially, we [13] used 150 interactions with a probability of answering by guessing of 0%, and in the previous section, we used 150 interactions with three different probabilities of answering correctly. However, in the present section, we now focus on identifying how large a data set is needed to find the size of the data set needed to find the characteristic curve of the model for this type of question that is accurate enough, similar to what has been done in other studies [42] [43]. To determine the characteristic curve of the model for this type of question, we simulated students' interactions in three exercises with different difficulties based on previous knowledge acquired: low, medium, and high. The data set size options for each exercise were as follows:

- 30 interactions
- 50 interactions
- 100 interactions
- 150 interactions
- 200 interactions
- 300 interactions
- 1000 interactions

C. Using Different Types of Distributions

In a previous work [13], a standard distribution was used for the simulations. However, in the present section, we performed simulations with different distributions for two exercise characteristics: grade, time spent, and the variable number of attempts to keep the distribution fixed in all simulations. The aim was to simulate different student behaviors and identify whether the model fits different possible real-world scenarios. Previous work has used different simulations, [44] used a uniform distribution for the difficulty of questions in simulated student interactions. On the other hand, [45] assumed student ability to be a normal distribution with mean and variance using it to obtain the probability of answering correctly in simulated students. The following distributions were used:

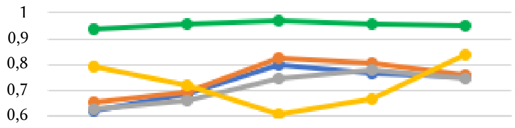
- Uniform distribution: Interactions are centered on intervals (a,b). A possible scenario is that students obtain a similar grade in an exercise, and none have low or high extremes.
- Normal distribution: These are the interactions used in the previous study and previous simulations; it is also the distribution used in other studies [45] [13] where student data were simulated.
- Gamma distribution: Most of the interactions were close to each other, and a few data points were at the end of the bell distribution. For example, almost all students had a similar grade in one exercise, and a few students had a higher grade.

V. Results

A. Using Different Probabilities of Answering Correctly

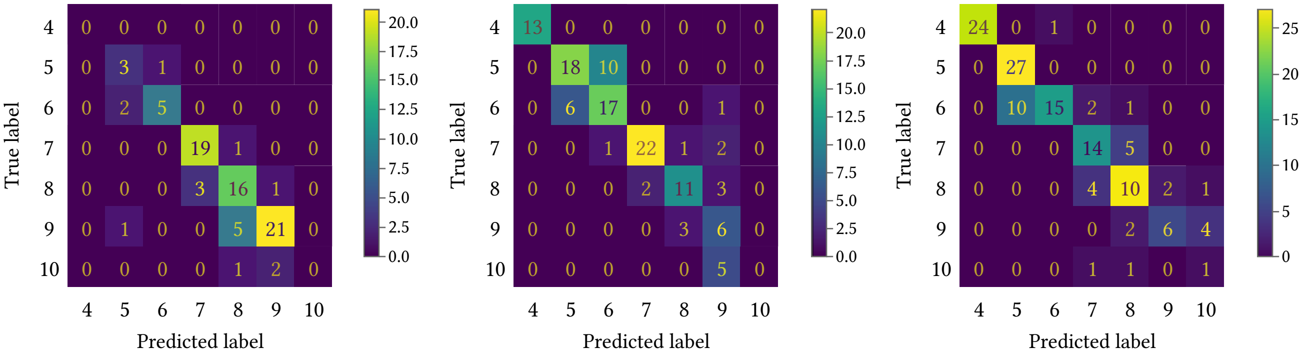
1. Curve Estimation Using Machine Learning

We tested the machine-learning algorithms described in Section III.B. 1 using the metrics previously indicated. As shown in Fig. 8, the best algorithm describing the three different datasets was the nearest neighbor with k equal to 10. The metrics corresponding to the 50% probability have poor results, with values between 0.6 and 0.7, owing to the high probability of answering correctly. Therefore, the model



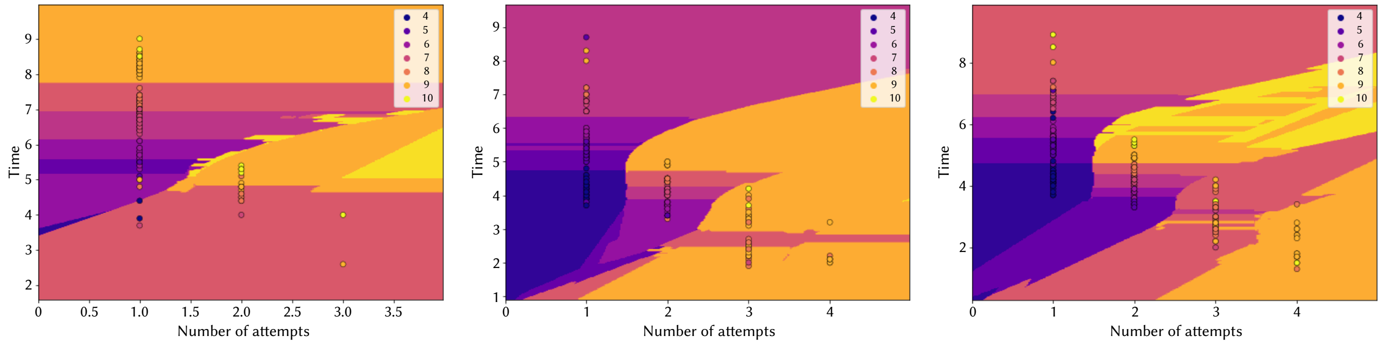
		50% probability	20% probability	14% probability	7% probability	5% probability
●	Precision (macro)	0,619	0,687	0,798	0,769	0,750
●	Recall (macro)	0,652	0,693	0,827	0,806	0,758
●	F1 (macro)	0,628	0,662	0,748	0,777	0,744
●	RMSE (macro)	0,795	0,722	0,610	0,668	0,837
●	AUC	0,940	0,960	0,970	0,960	0,950

Fig. 8. Metric of algorithm Nearest Neighbors (k=10).



(a) (b) (c)

Fig. 9. Confusion matrix with nearest neighbors (k=10).



(a) (b) (c)

Fig. 10. Plot with nearest neighbors(k=10).

had few classes corresponding to lower grades and could not learn correctly for these values. In this scenario, the proposed model design could not be used because it only had two answer choices with a maximum of two attempts to find the correct answer at random by the student. In this type of question with few possible answers, such as True/False, it is not recommended to use the model design because, after two attempts, all students would get the maximum grade, no matter how much time was spent.

In contrast, for probabilities of 5%, 7%, 14%, and 20%, there was a different distribution of grades and sufficient data obtained in each class of the dependent variable for learning the algorithm. By decreasing the probability of answering correctly, the values of all the metrics evaluated improved. The results were unsatisfactory with a probability of 20%, the results were unsatisfactory. In addition, for probabilities of 14%, 7%, and 5%, values greater than 0.75 are obtained in the precision, recall, and F1 metrics, respectively. In contrast, the AUC and RMSE metrics were relatively similar.

In the following subsections, we report the results for three probabilities of answering correctly: 50%, 14%, and 7%, and we select the probability for each type of question.

2. Best Algorithm Nearest Neighbors

Having already identified the best algorithm for predicting the grade, in this subsection, we present three different subsections with three different figures for each probability of answering correctly for the best algorithm. The three scenarios selected for analysis in this study and the following subsections are probabilities of 50%, 14%, and 7%. Fig. 9 shows the three confounding matrices for the algorithm in the three evaluated scenarios. Fig. 9(a) corresponds to a 50% probability of answering; it can be seen that the algorithm has few elements for grades better than 5 and has a test condition for grades 7 and 9. Also, Fig. 9(b) and Fig. 9(c) correspond to 14% and 7% probabilities respectively and Fig. 9(c) has a better accuracy between grades 4 and 7 and presents a particular sensitivity between grades greater than 7. A possible reason may be the small amount of data available for these classes.

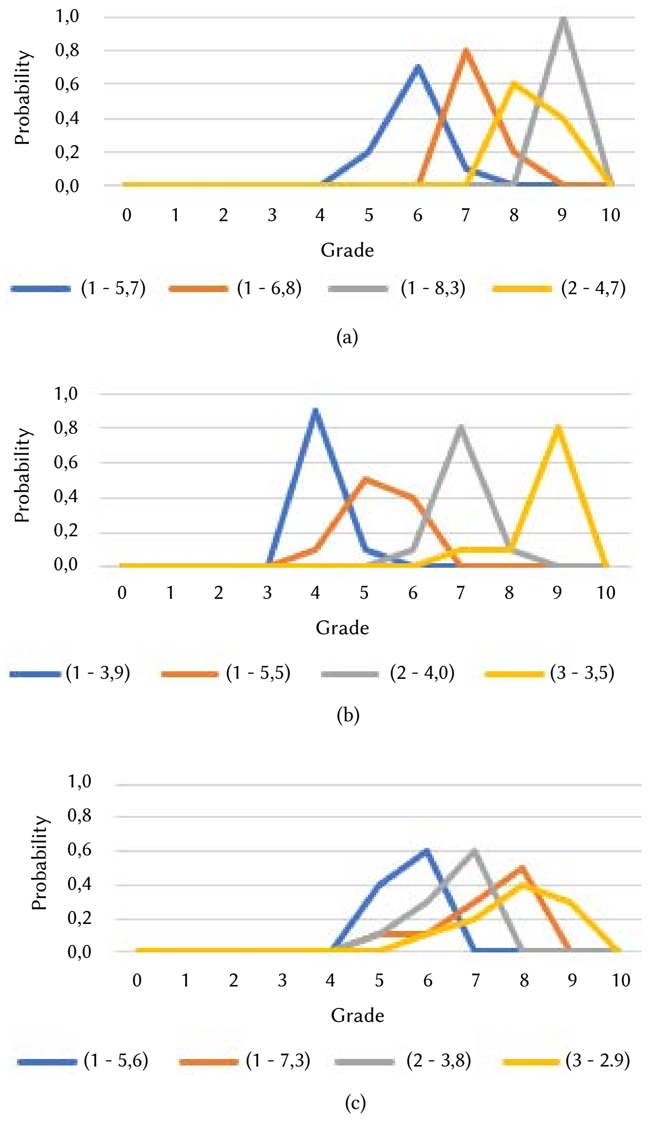
Fig. 10 shows the different clusters corresponding to the nearest neighbor algorithm with k equal to 10. The limits of each class vary depending on the percentage of probability of answering correctly. The tonality varies with color to yellow, corresponding to class 10 in the question with 50% (Fig. 10(a)), while 14% (Fig. 10(b)) and 7% (Fig. 10(c)) show the whole range of tonality from the blue of class 0 to the yellow color corresponding to class 10.

To avoid overfitting, we evaluated the score of the algorithm using the *cross_val_score* of the sklearn.model_selection library with accuracy as scoring and 5-fold cross-validation. Table I shows the results of the cross-validation performed with five subsets of the nearest neighbors algorithm with the three different datasets representing the three types of questions. The N-fold column indicates the run number and the value indicates the accuracy of the algorithm in this run. Thus, we evaluated the robustness of the algorithm and avoided overfitting.

TABLE I. Cross-Validation Values

	50% probability	14% probability	7% probability
cv1	0,679	0,799	0,768
cv2	0,575	0,826	0,793
cv3	0,616	0,812	0,692
cv4	0,676	0,740	0,781
cv5	0,548	0,809	0,806

Finally, we used the *predict_proba* method to calculate the probability of different grades using the simulated test dataset with three different probabilities of answering correctly. For example, in Fig. 11, using in the model a similar ordered pair, such as (1–5.5), (1–5.6), or (1–5.7), where 1 means the number of attempts and 5.5,5.6,5.7



(a)

(b)

(c)

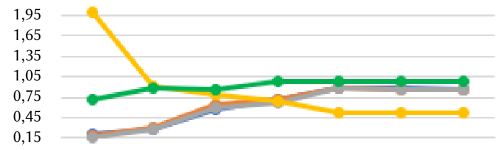
Fig. 11. Predicted probability with nearest neighbors(k=10).

corresponds to the time spent in the exercise. We obtained a probability of 0.7 for a grade of 6 in the exercise with 50%correct answers (Fig. 11(a)). In comparison, with an exercise of 14% (Fig. 11(b)), we obtained a probability of 0.9 for a grade of 4, and an exercise of 7% (Fig. 11(c)), we obtained a 0.6 probability for a grade of 6. As we can observe, we obtained different probabilities in the classes for the three probabilities of answering correctly, evaluated with similar values of time spent in the first attempt.

B. Using Different Number of Interactions

1. Curve Estimation Using Machine Learning

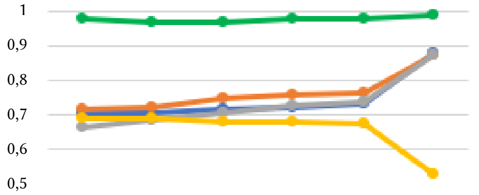
We tested the same algorithm used in the previous section, using the same metrics. Fig. 12 shows the precision, recall, F1, RMSE, and AUC metrics. We can see an increase in their values as the number of interactions increased, stabilizing the curve at 200 interactions. The RMSE metric decreased as the number of interactions increased, achieving stability with the same number of interactions as that of the other metrics. From the results, we can conclude that the minimum number of interactions for the proposed exercise model with good accuracy is approximately 200 because the best results were obtained for all the metrics evaluated: precision of 0.878, recall of 0.873, F1 of 0.875, and RMSE of 0.527.



		30 interactions	50 interactions	100 interactions	150 interactions	200 interactions	300 interactions	1000 interactions
●	Precision (macro)	0,214	0,269	0,568	0,708	0,878	0,882	0,866
●	Recall (macro)	0,19	0,315	0,632	0,718	0,873	0,862	0,871
●	F1 (macro)	0,167	0,281	0,587	0,666	0,875	0,865	0,864
●	RMSE (macro)	2,000	0,918	0,793	0,693	0,527	0,526	0,527
●	AUC	0,723	0,890	0,870	0,980	0,990	0,985	0,989

Fig. 12. Metric of algorithm Nearest Neighbors (k=10).

Regarding the previous result, we performed simulations with values between 150 and 200 interactions to determine the exact value at which the curve of the metrics stabilizes. Fig. 13 shows the results of all the metrics evaluated, and the results show that between the values of 150 and 190 interactions, the values of the metrics precision, recall, F1, and RMSE increase slightly. For the 200 interactions, all metric values yielded the best results, as indicated in the previous paragraph. In summary, for the simulated exercise with values for the three characteristics, the number of attempts (mean, 2; minimum, 1; maximum, 4) of 200 interactions was needed.

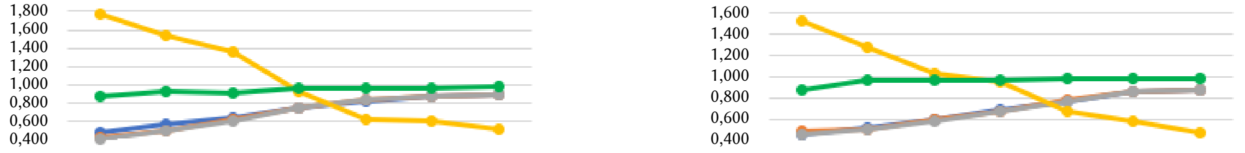


		150 interactions	160 interactions	170 interactions	180 interactions	190 interactions	200 interactions
●	Precision (macro)	0,708	0,709	0,716	0,721	0,731	0,878
●	Recall (macro)	0,718	0,725	0,747	0,757	0,764	0,873
●	F1 (macro)	0,666	0,687	0,706	0,727	0,737	0,875
●	RMSE (macro)	0,693	0,692	0,683	0,683	0,674	0,527
●	AUC	0,98	0,970	0,970	0,980	0,980	0,990

Fig. 13. Metric of algorithm Nearest Neighbors (k=10) between 150 and 200.

If the number of attempts to obtain the maximum grade increases, then. To test this hypothesis, we simulated two new exercises by modifying the variable number of attempts differently from the previous exercise with a mean of two. The first exercise involved an average of four attempts, and the second exercise involved an average of seven attempts.

Fig. 14(a) shows the results of the precision, recall, F1, RMSE, and AUC metrics for different interactions in the exercise with a mean of four attempts. The metrics decreased as the number of interactions increased, except for the AUC metric, which tended to maintain similar values. Fig. 14 shows that in 500 interactions, good values were obtained for all metrics evaluated: *precision* = 0.815, *recall* = 0.83, *F1* = 0.84, *RMSE* = 0.628, and *AUC* = 0.92. We conclude that, for an exercise with a mean of four attempts to find the maximum grade, a minimum of 500 interactions are needed.



		30 interactions	50 interactions	100 interactions	150 interactions	200 interactions	300 interactions	1000 interactions
●	Precision (macro)	0,480	0,559	0,645	0,748	0,815	0,880	0,894
●	Recall (macro)	0,425	0,495	0,629	0,754	0,830	0,876	0,889
●	F1 (macro)	0,412	0,493	0,612	0,752	0,840	0,877	0,891
●	RMSE (macro)	1,770	1,535	1,360	0,927	0,628	0,596	0,509
●	AUC	0,880	0,920	0,910	0,960	0,970	0,970	0,980

(a)

		30 interactions	50 interactions	100 interactions	150 interactions	200 interactions	300 interactions	1000 interactions
●	Precision (macro)	0,459	0,509	0,595	0,681	0,770	0,857	0,881
●	Recall (macro)	0,490	0,496	0,588	0,669	0,774	0,858	0,879
●	F1 (macro)	0,455	0,495	0,583	0,670	0,771	0,857	0,880
●	RMSE (macro)	1,531	1,283	1,034	0,949	0,673	0,581	0,469
●	AUC	0,870	0,960	0,970	0,970	0,990	0,980	0,990

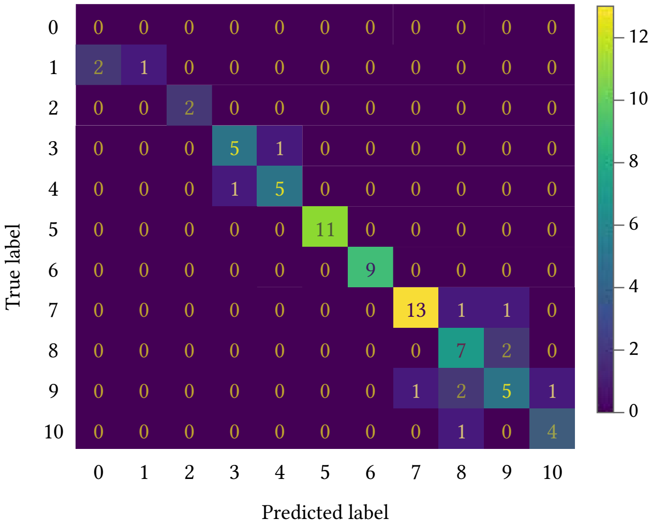
(b)

Fig. 14. Metric of algorithm Nearest Neighbors (k=10) increasing the number of attempts.

Furthermore, Fig. 14 (b) shows the results of the same metrics in the exercise with a mean of seven attempts. In the exercise with a mean of four attempts, the metrics decreased as the number of interactions increased, except for the AUC metric. Fig. 14(b) shows that at 800 interactions, good values were obtained for all metrics: *precision* = 0.875, *recall* = 0.858, *F1* = 0.857, *RMSE* = 0.581, and *AUC* = 0.980. Based on these results, we can conclude that, in an exercise with a mean of seven attempts, a minimum of 800 interactions would be needed. In summary, as the number of attempts that students must make to obtain the maximum grade increases, the minimum number of simulated students also increases.

2. Best Algorithm Nearest Neighbors (K=10) With 200 Interactions

Fig. 15 shows the confusion matrix of the algorithm with the highest accuracy with 200 interactions obtained in the previous subsection. The results show that the algorithm has a better prediction for middle grades, with a decreasing prediction as the grades increase. However, for low grades, the accuracy decreases because of the small dataset with which the model was trained and because of the type of distribution used in the simulations.



	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	2	1	0	0	0	0	0	0	0	0	0
2	0	0	2	0	0	0	0	0	0	0	0
3	0	0	0	5	1	0	0	0	0	0	0
4	0	0	0	1	5	0	0	0	0	0	0
5	0	0	0	0	0	11	0	0	0	0	0
6	0	0	0	0	0	0	9	0	0	0	0
7	0	0	0	0	0	0	0	13	1	1	0
8	0	0	0	0	0	0	0	0	7	2	0
9	0	0	0	0	0	0	0	1	2	5	1
10	0	0	0	0	0	0	0	0	1	0	4

Fig. 15. Confusion matrix with nearest neighbors(k=10).

Additionally, Fig. 16 shows different clusters of the algorithm for different numbers of attempts. The tonalities of the different classes varied as number of attempts increased, the students improved their grades, and the total number of classes decreased. In the middle grade, most of the clusters were located in the correct class on the first attempt. In the low and high grades, the classes had clusters corresponding to the nearby classes.

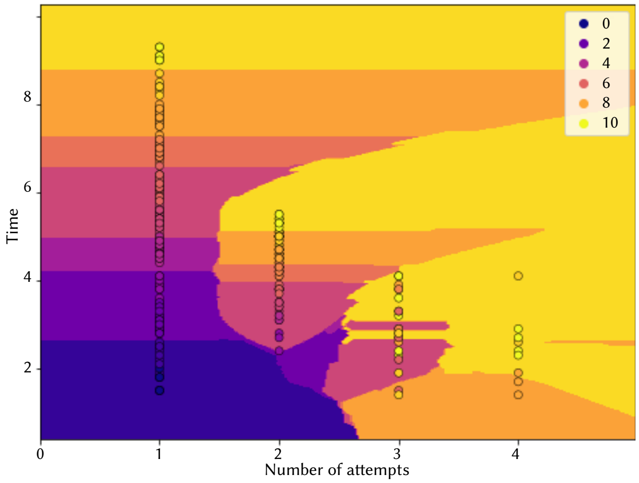


Fig. 16. Plot with nearest neighbors(k=10).

Using the same method as in Section III.A.2, Table II lists the variables considered in the model with their possible values. The N-fold column indicates the run number and the value indicates the accuracy of the algorithm in this run. The results show good accuracy of the algorithm in the five different subsets of the cross-validation, indicating the robustness of the algorithm and the avoidance of overfitting.

TABLE II. Cross-Validation Values

N-fold	value
cv1	0,892
cv2	0,965
cv3	0,862
cv4	0,789
cv5	0,862

Finally, we used the *predict_proba* method to calculate the probability of different grades using the simulated test data. Fig. 17 shows the results obtained using these four examples. The first example (1-3.0) corresponds to the first attempt in a time of 3 units, which corresponds to a higher probability of obtaining a grade of 2. In contrast, (3–3.1) corresponds to the third attempt at a time of 3.1. Similar to the previous study, the results vary, obtaining a higher probability for a grade of 7 or 10. In conclusion, according to the predictions of this exercise, if a student spends more time during the first attempt or makes more than one attempt, the student has a greater probability of obtaining a higher grade.

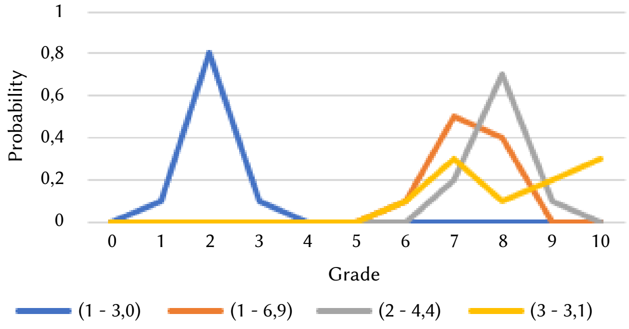
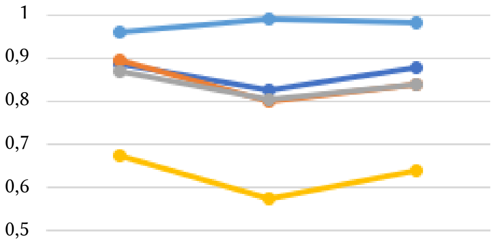


Fig. 17. Predicted probability with nearest neighbors(k=10).

C. Using Different Types of Distributions

1. Curve Estimation Using Machine Learning

To answer RQ4, we trained the model using the three datasets corresponding to each distribution. The results were obtained using the same algorithm as that in the previous section, and the same metrics are shown in Fig. 18. In general, for the three different student behaviors (three distributions), the precision, recall, and F1 metrics exhibited values ranging from 0,8 and 0,9. In contrast, the RMSE decreased slightly when the student's behavior was normally distributed. Finally, the AUC of the three distributions was not significantly different, with a value very close to 1. In conclusion, the different distributions of student behavior using machine learning algorithms converged with good results.

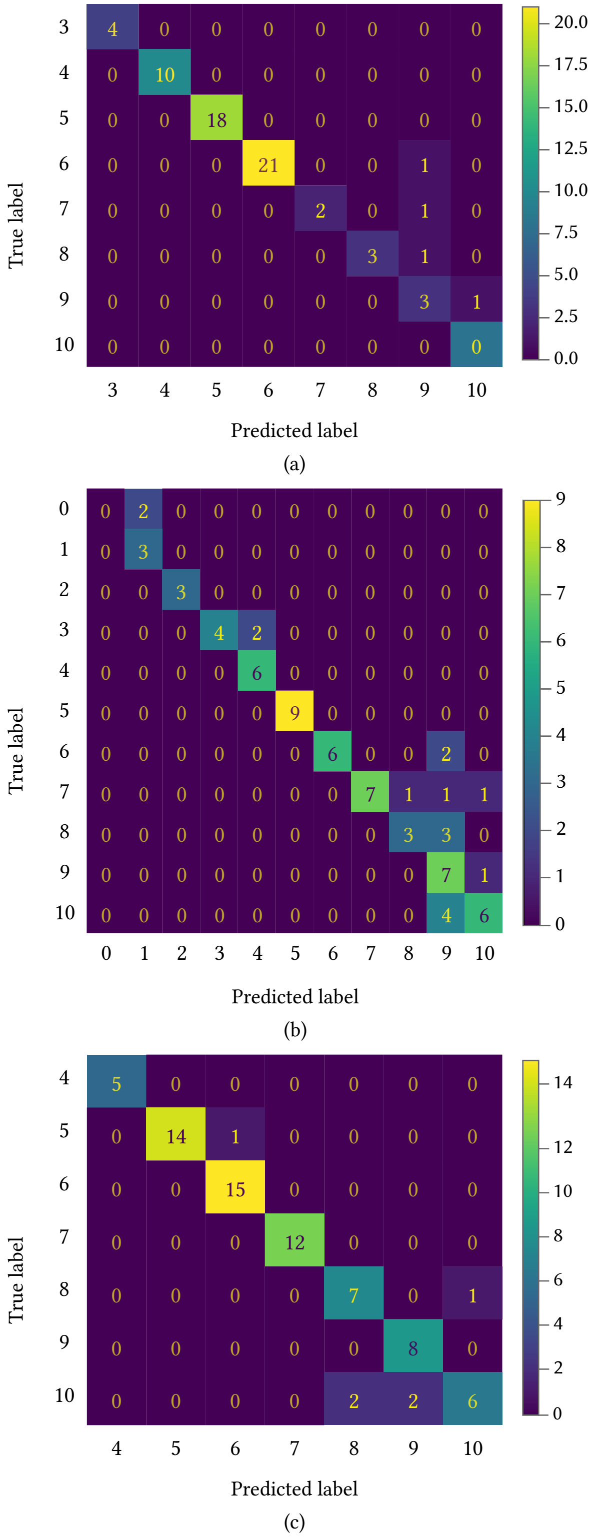


		Uniform distribution	Normal distribution	Gamma distribution
●	Precision (macro)	0,885	0,827	0,875
●	Recall (macro)	0,896	0,801	0,840
●	F1 (macro)	0,869	0,804	0,836
●	RMSE (macro)	0,672	0,574	0,639
●	AUC	0,96	0,990	0,980

Fig. 18. Metric of algorithm Nearest Neighbors (k=10).

2. Best Algorithm Nearest Neighbors (K=10)

The confusion matrix allowed us to observe the behavior of the algorithm by relating the preconditions to the real cases. Fig. 19 shows the confusion matrices for the three different distributions used. Fig. 19(a) corresponds to a uniform distribution; Fig. 19(b) corresponds to a normal distribution; and Fig. 19(c) corresponds to a gamma distribution. Fig. 19(a) and Fig. 19(c) show that not all values are available for the degree of the dependent variable. By contrast, in Fig. 19(b), all grade classes can be obtained. Moreover, Fig. 19(a) and Fig. 19(c) show similar behavior, obtaining a high precision for the mean grades, whereas Fig. 19(b) shows good precision distributed over a larger number of grades.



True label \ Predicted label	3	4	5	6	7	8	9	10
3	4	0	0	0	0	0	0	0
4	0	10	0	0	0	0	0	0
5	0	0	18	0	0	0	0	0
6	0	0	0	21	0	0	1	0
7	0	0	0	0	2	0	1	0
8	0	0	0	0	0	3	1	0
9	0	0	0	0	0	0	3	1
10	0	0	0	0	0	0	0	0

(a)

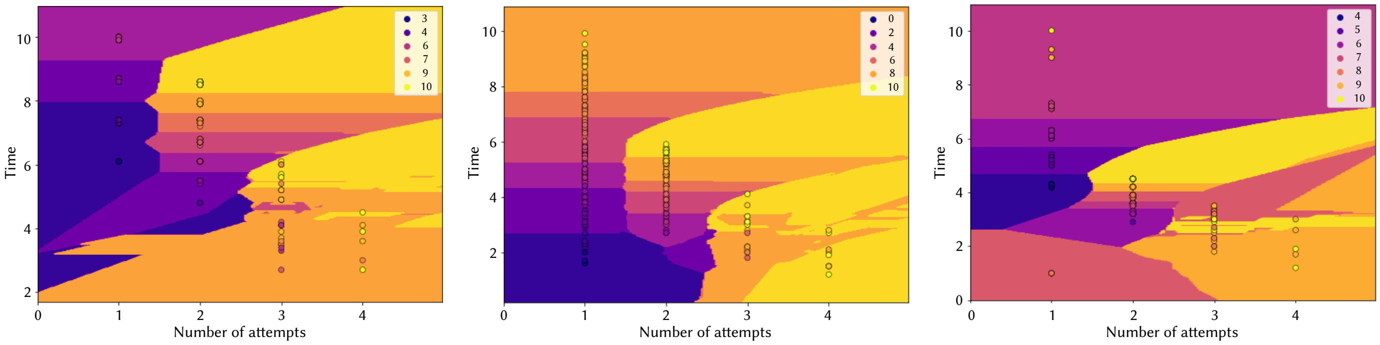
True label \ Predicted label	0	1	2	3	4	5	6	7	8	9	10
0	0	2	0	0	0	0	0	0	0	0	0
1	0	3	0	0	0	0	0	0	0	0	0
2	0	0	3	0	0	0	0	0	0	0	0
3	0	0	0	4	2	0	0	0	0	0	0
4	0	0	0	0	6	0	0	0	0	0	0
5	0	0	0	0	0	9	0	0	0	0	0
6	0	0	0	0	0	0	6	0	0	2	0
7	0	0	0	0	0	0	0	7	1	1	1
8	0	0	0	0	0	0	0	0	3	3	0
9	0	0	0	0	0	0	0	0	0	7	1
10	0	0	0	0	0	0	0	0	0	4	6

(b)

True label \ Predicted label	4	5	6	7	8	9	10
4	5	0	0	0	0	0	0
5	0	14	1	0	0	0	0
6	0	0	15	0	0	0	0
7	0	0	0	12	0	0	0
8	0	0	0	0	7	0	1
9	0	0	0	0	0	8	0
10	0	0	0	0	2	2	6

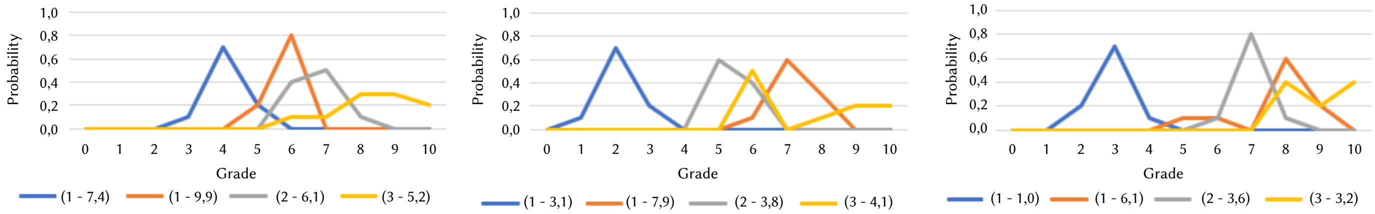
(c)

Fig. 19. Confunsion matrix with nearest neighbors(k=10).



(a) (b) (c)

Fig. 20. Plot with nearest neighbors(k=10).



(a) (b) (c)

Fig. 21. Predicted probability with nearest neighbors(k=10).

Next, Fig. 20 shows the different clusters of the dependent variable using the nearest neighbor algorithm with k equal to 10. Fig. 20(a) corresponds to the uniform distribution, Fig. 20(b) a normal distribution and Fig. 20 for the gamma distribution. The three figures represent three different student behaviors, where the different shades represent each class of the grade variable. The dispersion of the clusters in the different attempts was related to the type of distribution used, as shown in Fig. 20(a) and Fig. 20(c), with a large dispersion in each attempt. In addition, Fig. 20(b) shows a better distribution of clusters in each class, as represented by the same colors.

Using the same method as in Section III.A.2, Table III presents the results of the cross-validation performed with three subsets representing the three types of distributions. The N-fold column indicates the run number and the value indicates the accuracy of the algorithm in this run. Accuracy of different subsets in cross-validation obtained good results, demonstrating the robustness of the algorithm in different distributions and avoiding overfitting of the algorithm.

TABLE III. Cross-Validation Values

N-fold	Uniform distribution	Normal distribution	Gamma distribution
cv1	0,849	0,824	0,887
cv2	0,876	0,810	0,928
cv3	0,917	0,838	0,914
cv4	0,903	0,796	0,818
cv5	0,876	0,867	0,832

Finally, we used the *predict_prob method* to calculate the probability of different grades, and the dataset used as a test was a part of the simulated data. Fig. 21 shows the results obtained using three different simulations. Fig. 21(a) corresponds to a uniform distribution, Fig. 21(b) corresponds to a normal distribution, and Fig. 21(c) corresponds to a gamma distribution. In conclusion, the probabilities obtained had different values for the three evaluated datasets.

VI. Discussion

In this section, we analyze the results obtained from the simulations based on our research questions.

A. RQ1: Is It Possible to Obtain a Time-Grade-Attempts Model in Exercises That Are Accurate Enough Using Some Traditional Machine Learning Algorithms?

The results show that traditional machine learning algorithms can model exercises using independent variables, such as time and number of attempts, with the dependent variable being the grade obtained by the student. The four algorithms that could be used were nearest neighbor (k=10), Decision Tree (Max Depth=10), Random Forest (Max Depth=10), and MLP (tanh). We evaluated the effectiveness of all algorithms based on metrics such as precision, recall, F1, RMSE, and AUC and found that these four algorithms yielded the best results. Finally, we recommend using the nearest neighbor algorithm (k=10) because the first choice, as it achieved the best results in the simulations conducted. This algorithm and its variants have been used in various applications such as medical predictions, data mining, and financial modeling [46].

B. RQ2: How Does the Accuracy of the Models Vary for Different Types of Exercises?

The findings show a variation in the values of the metrics evaluated for the different questions. Modifying the probability of answering correctly implies obtaining different data dispersions among dependent variable classes (grades). First, the questions with a 50% probability had the worst result among the others. This is because of the high probability of obtaining a good grade randomly even if the student has no prior knowledge. As the probability of answering correctly decreases, better results are obtained in the metrics because of the data distribution, and the algorithm has the necessary information to learn correctly. We do not recommend using the proposed model for true/false questions corresponding to 50% probability because of its low effectiveness and the limited data that will be obtained regarding the number of student attempts.

By contrast, the model has better results in the metrics using an artificial intelligence algorithm for the type of question with multiple options represented by probabilities of 14%, 7%, and 5%. For 20% probability, the results show a small increase in the results for 50% probability. There is an inverse relationship between probability and metrics; as the probability of answering correctly increases, the values of the evaluated metrics decrease.

This type of question is perceived as better and preferred by the students [47]. Using the information obtained from the model, teachers can orchestrate the process by redesigning educational exercises to improve student learning, as in other studies [48] [49]. For example, by knowing the types of questions, teachers can modify their exams based on student's grades, the number of attempts that students will have to make, and the time it will take to finish the questions. By using this information, teachers can redesign questions with better results based on the proposed model to improve students' learning processes.

C. RQ3: What Minimum Number of Interactions Is Required to Stabilize the Exercise Model With Acceptable Accuracy?

Previous studies [42] [43] examined the impact of various sample sizes on model stability and accuracy, to identify the minimum number of sizes required to optimize the characteristic curve. The results show that we need a minimum of 200 student interactions in the exercise to model the three characteristics of the proposed model design for exercises, with an average of two attempts to obtain the maximum grade.

However, we could also use 300 or more interactions for the first attempt because the difference in accuracy was insignificant because there were few classes to classify. The accuracy of the algorithm did not increase significantly in the model considering the threshold of 200 interactions as the number of interactions increased. Having a minimum of 200 interactions performed in an exercise does not necessarily imply having 200 students because the same student can perform multiple interactions when trying to solve the same exercise several times, which increases the total number of interactions.

Moreover, if the exercises require more attempts to obtain the maximum grade, such as 4 or 7, more interactions are required to converge the model. The findings showed that we would need a minimum of 500 and 800 interactions for these two types of exercises. Existing a direct relation between the number of attempts and the number of interjections, if the number of attempts needed to obtain the maximum score increases, the number of interactions will be higher.

The results can be used to analyze any platform on which the proposed exercise model should be tested: for example, in a massive open online course (MOOC), because of the large number of learners and the possibility of obtaining numerous interactions; or in contrast, in Learning Management System (LMS) courses with a specific number of learners.

D. RQ4: How Do Different Forms of Student Behavior Modify the Results Obtained in the Previous Research Questions?

We used different distributions in studies of students with exercise characteristics. For example, normal distributions have been used to address question difficulty and student skills [50]. Once we identified the number of interactions and type of questions required to estimate the model and obtained good results for all metrics, we ran simulations with different distributions to recreate possible student scenarios. For example, students obtained an average grade on the first attempt; however, after several attempts, they could not improve their grades. No matter how many attempts the students made, they could not get the maximum grade, or all students achieved high grades on their first attempt. The findings allow us to infer that we can adapt the proposed model to different scenarios to help teachers identify the problems that students face when solving the exercise and redesign the exercise if necessary to improve the grade obtained by the student.

VII. Conclusion and Future Work

The simulations allowed us to illustrate the exercise model under different scenarios. First, we found that a traditional machine learning algorithm could model the exercise while obtaining acceptable results for the metrics evaluated, and the robustness of the model was evaluated using five-field cross-validation. Next, we found that different probabilities of answering a question correctly affect the accuracy of the model due to the distribution of the scores obtained as a function of the probability; for questions with few answer options, we do not recommend using the proposed model design as in the case of a true/false question.

Moreover, identifying the number of interactions is essential for testing the model because it indicates the minimum number of students required to evaluate the model accurately. The findings indicated that the number of interactions is related to the number of attempts required to obtain the maximum grade. For example, model design requires a minimum of 200 interactions for an exercise, with an average of two attempts. However, if the number of attempts is increased, more interactions will be required to converge the model.

Also, the model design converges on the different interaction scenarios of the students simulating different student behaviors using three different distributions so that future work can evaluate the model with other distributions.

The present study's findings will allow teachers to redesign their course exercises, knowing information about the three characteristics of exercise: number of attempts, time, and grade for each type of question. For example, identifying student patterns in exercises, such as students failing to improve their grades after a few attempts or investing too much time to improve their grades, among others.

A limitation of the present study is the use of simulated data rather than real data, which does not allow us to generalize the accuracy obtained by applying the different algorithms tested for the three exercise characteristics. However, the results obtained allowed us to illustrate the behavior of the model in the different scenarios evaluated and identify the features needed in the test design in a real environment.

In future work, we plan to test the model in a real-world scenario based on simulation results. This would require many students to interact with the educational exercise. Hence, a massive course is necessary to evaluate the generality of the proposed model. Additionally, we will create explainable visualizations with information about the model within the dataset, that the teacher can use to detect exercise patterns using the three indicated characteristics. Moreover, the teacher can redesign the exercise based on these visualizations to enhance the students' learning.

Acknowledgment

This work was supported in part by the FEDER/Ministerio de Ciencia, Innovación y Universidades–Agencia Estatal de Investigación, through the Smartlet Project under Grant TIN2017-85179-C3-1-R and the H2O Learn Project under Grant PID2020-112584 RB-C31, in part by the Madrid Regional Government through the e-Madrid-CM Project under Grant S2018/TCS-4307 a project which is co-funded by the European Structural Funds (FSE and FEDER).

References

[1] P. Brusilovsky, S. Edwards, A. Kumar, L. Malmi, L. Benotti, D. Buck, P. Ihantola, R. Prince, T. Sirkiä, S. Sosnovsky, *et al.*, "Increasing adoption of smart learning content for computer science education," in *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference*, 2014, pp. 31–57.

[2] E. G. Rincon-Flores, E. Lopez-Camacho, J. Mena, O. Olmos, "Teaching through learning analytics: Predicting student learning profiles in a physics course at a higher education institution," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 7, pp. 82–89, 2022.

[3] M. O. Edelen, B. B. Reeve, "Applying item response theory (irt) modeling to questionnaire development, evaluation, and refinement," *Quality of Life Research*, vol. 16, no. 1, pp. 5–18, 2007.

[4] I. Rushkin, I. Chuang, D. Tingley, "Modelling and using response times in online courses," *arXiv preprint arXiv:1801.07618*, 2018.

[5] A. Jiménez-Macías, P. J. Muñoz-Merino, M. Ortiz- Rojas, M. Muñoz-Organero, C. Delgado Kloos, "Content modeling in smart learning environments: A systematic literature review," *Journal of Universal Computer Science (JUCS)*, vol. 30, no. 3, pp. 333–362, 2024.

[6] P. M. Moreno-Marcos, D. M. de la Torre, G. G. Castro, P. J. Muñoz-Merino, C. D. Kloos, "Should we consider efficiency and constancy for adaptation in intelligent tutoring systems?," in *International Conference on Intelligent Tutoring Systems*, 2020, pp. 237–247, Springer.

[7] M. Feng, J. Beck, N. Heffernan, K. Koedinger, "Can an intelligent tutoring system predict math proficiency as well as a standardized test?," in *Proceedings of the 1st International Conference on Education Data Mining*, 2008, pp. 107–116.

[8] B. Martin, A. Mitrovic, K. R. Koedinger, S. Mathan, "Evaluating and improving adaptive educational systems with learning curves," *User Modeling and User-Adapted Interaction*, vol. 21, pp. 249–283, 2011.

[9] F. Dorça, "Implementation and use of simulated students for test and validation of new adaptive educational systems: A practical insight," *International Journal of Artificial Intelligence in Education*, vol. 25, no. 3, pp. 319–345, 2015.

[10] E. Poitras, Z. Mayne, L. Huang, T. Doleck, L. Udy, S. Lajoie, "Simulated student behaviors with intelligent tutoring systems: Applications for authoring and evaluating network-based tutors," *Tutoring and Intelligent Tutoring Systems. Nova Publishers*, 2018.

[11] J. Champaign, R. Cohen, "A model for content sequencing in intelligent tutoring systems based on the ecological approach and its validation through simulated students," in *Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference (FLAIRS 2010)*, 2010, pp. 486–491.

[12] R. Pelánek, "Metrics for evaluation of student models.," *Journal of Educational Data Mining*, vol. 7, no. 2, pp. 1–19, 2015.

[13] A. Jiménez-Macías, P. J. Muñoz-Merino, C. Delgado Kloos, "A model to characterize exercises using probabilistic methods," in *Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'21)*, 2021, pp. 594–599.

[14] J. M. Spector, "Smart learning environments: Concepts and issues," Society for Information Technology & teacher education international conference, 2016, pp. 2728–2737, Association for the Advancement of Computing in Education (AACE).

[15] E. Pecheanu, C. Segal, D. Stefanescu, "Content modeling in intelligent instructional environments," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 2003, pp. 1229–1234, Springer.

[16] J. P. Lalor, H. Wu, H. Yu, "Learning latent parameters without human response patterns: Item response theory with artificial crowds," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, vol. 2019, 2019, p. 4240, NIH Public Access.

[17] F. Martínez-Plumed, R. B. Prudêncio, A. Martínez- Usó, J. Hernández-Orallo, "Item response theory in ai: Analysing machine learning classifiers at the instance level," *Artificial Intelligence*, vol. 271, pp. 18–42, 2019.

[18] C.-M. Chen, H.-M. Lee, Y.-H. Chen, "Personalized e-learning system using item response theory," *Computers & Education*, vol. 44, no. 3, pp. 237–255, 2005.

[19] D. Abbakumov, "The solution of the "cold start problem" in e-learning," *Procedia-Social and Behavioral Sciences*, vol. 112, pp. 1225–1231, 2014.

[20] K. Xue, V. Yaneva, C. Runyon, P. Baldwin, "Predicting the difficulty and response time of multiple choice questions using transfer learning," in *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2020, pp. 193–197.

[21] V. Yaneva, P. Baldwin, J. Mee, *et al.*, "Predicting the difficulty of multiple choice questions in a high- stakes medical exam," Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, 2019, pp. 11–20.

[22] Z. Qiu, X. Wu, W. Fan, "Question difficulty prediction for multiple choice problems in medical exams," Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 139–148.

[23] L. Benedetto, A. Cappelli, R. Turrin, P. Cremonesi, "R2de: a nlp approach to estimating irt parameters of newly generated questions," in *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, 2020, pp. 412–421.

[24] B. A. Lehman, D. Zapata-Rivera, "Student emotions in conversation-based assessments," *IEEE Transactions on Learning Technologies*, vol. 11, no. 1, pp. 41–53, 2018.

[25] N. Capuano, S. Caballé, J. Conesa, A. Greco, "Attention-based hierarchical recurrent neural networks for mooc forum posts analysis," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13, 2020.

[26] T. Atapattu, K. Falkner, M. Thilakaratne, L. Sivaneasharajah, R. Jayashanka, "What do linguistic expressions tell us about learners' confusion? a domain-independent analysis in moocs," *IEEE Transactions on Learning Technologies*, vol. 13, no. 4, pp. 878–888, 2020.

[27] M. Feng, N. Heffernan, K. Koedinger, "Addressing the assessment challenge with an online system that tutors as it assesses," *User modeling and user-adapted interaction*, vol. 19, no. 3, pp. 243–266, 2009.

[28] E. Verdú, M. J. Verdú, L. M. Regueras, J. P. de Castro, R. García, "A genetic fuzzy expert system for automatic question classification in a competitive learning environment," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7471–7478, 2012.

[29] M. Uto, "Rater-effect irt model integrating supervised lda for accurate measurement of essay writing ability," in *International Conference on Artificial Intelligence in Education*, 2019, pp. 494–506, Springer.

[30] H. A.-M. Gerlache, P. M. Ger, L. de la Fuente Valentín, "Towards the grade's prediction. a study of different machine learning approaches to predict grades from student interaction data," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 4, pp. 196–204, 2022.

[31] K. VanLehn, S. Ohlsson, R. Nason, "Applications of simulated students: An exploration," *Journal of artificial intelligence in education*, vol. 5, pp. 135–135, 1994.

[32] N. Matsuda, W. W. Cohen, K. R. Koedinger, "Teaching the teacher: tutoring simstudent leads to more effective cognitive tutor authoring," *International Journal of Artificial Intelligence in Education*, vol. 25, no. 1, pp. 1– 34, 2015.

[33] S. B. Blessing, "A programming by demonstration authoring tool for model-tracing tutors," *International Journal of Artificial Intelligence in Education*, vol. 8, no. 3- 4, pp. 233–261, 1997.

[34] D. E. K. Lelei, G. McCalla, "Simulation in support of lifelong learning design: A prospectus.," in *SLLL@ AIED*, 2019, pp. 38–42.

[35] A. C. Graesser, "Conversations with autotutor help students learn," *International Journal of Artificial Intelligence in Education*, vol. 26, no. 1, pp. 124–132, 2016.

[36] A. Vizcaíno, "A simulated student can improve collaborative learning," *International Journal of Artificial Intelligence in Education*, vol. 15, no. 1, pp. 3–40, 2005.

[37] D. E. K. Lelei, G. McCalla, "How many times should a pedagogical agent simulation model be run?," in *International Conference on Artificial Intelligence in Education*, 2019, pp. 182–193, Springer.

[38] G. Erickson, S. Frost, S. Bateman, G. McCalla, "Using the ecological approach to create simulations of learning environments," in *Artificial Intelligence in Education*, 2013, pp. 411–420, Springer.

[39] S. Frost, G. McCalla, "Exploring through simulation an instructional planner for dynamic open-ended learning environments," in *Artificial Intelligence in Education*, 2015, pp. 578–581, Springer.

[40] M. A. Riedesel, N. Zimmerman, R. Baker, T. Titchener, J. Cooper, "Using a model for learning and memory to simulate learner response in spaced practice," in *Artificial Intelligence in Education*, 2017, pp. 644–649,

Springer.

[41] M. Dzikovska, N. Steinhauser, E. Farrow, J. Moore, G. Campbell, "Beetle ii: Deep natural language understanding and automatic feedback generation for intelligent tutoring in basic electricity and electronics," *International Journal of Artificial Intelligence in Education*, vol. 24, pp. 284–334, 2014.

[42] T. R. O'Neill, J. L. Gregg, M. R. Peabody, "Effect of sample size on common item equating using the dichotomous rasch model," *Applied Measurement in Education*, vol. 33, no. 1, pp. 10–23, 2020.

[43] Q. He, C. Wheadon, "The effect of sample size on item parameter estimation for the partial credit model," *International Journal of Quantitative Research in Education*, vol. 1, no. 3, pp. 297–315, 2013.

[44] M. Antal, "On the use of elo rating for adaptive assessment," *Studia Universitatis Babes-Bolyai, Informatica*, vol. 58, no. 1, pp. 29–41, 2013.

[45] R. Pelánek, J. Rihák, J. Papoušek, "Impact of data collection on interpretation and evaluation of student models," in *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, 2016, pp. 40–47.

[46] K. Taunk, S. De, S. Verma, A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 2019, pp. 1255–1260, IEEE.

[47] K. Struyven, F. Dochy, S. Janssens, "Students' perceptions about evaluation and assessment in higher education: A review," *Assessment & Evaluation in Higher Education*, vol. 30, no. 4, pp. 325–341, 2005.

[48] M. F. Rodríguez, J. Hernández Correa, M. Pérez- Sanagustín, J. A. Pertuze, C. Alario-Hoyos, "A mooc-based flipped class: Lessons learned from the orchestration perspective," in *European Conference on Massive Open Online Courses*, 2017, pp. 102–112, Springer.

[49] L. P. Prieto, Y. Dimitriadis, J. I. Asensio-Pérez, C.-K. Looi, "Orchestration in learning technology research: evaluation of a conceptual framework," *Research in Learning Technology*, vol. 23, 2015.

[50] J. Niznan, J. Papousek, R. Pelánek, "Exploring the role of small differences in predictive accuracy using simulated data," in *AIED Workshop Proceedings*, vol. 5, 2015, pp. 21–30.

Alberto Jiménez-Macías

Alberto Jiménez-Macías is a PhD student at Universidad Carlos III de Madrid. He obtained a bachelor's degree in Telematics Engineering and a master's degree in Computer Science at the Escuela Superior Politécnica del Litoral (ESPOL) (Ecuador). He carried out development and research work at the Information Technology Center (CTI-ESPOL) for 8 years. His areas of interest are Learning Analytics, Educational Data Mining and Educational Technology.

Pedro J. Muñoz-Merino

Pedro J. Muñoz-Merino is Full Professor at the Department of Telematics Engineering at Universidad Carlos III de Madrid. In 2003, he received his Telecommunication Engineering degree from the Polytechnic University of Valencia, and in 2009 his PhD in Telematics Engineering from the Universidad Carlos III de Madrid. He has been the coordinator of the LALA project, a project funded by the European Commission for the adoption of learning analytics in Latin America. He has also participated in more than 40 research projects at the international and national level, also including several contracts with companies, being the Principal Investigator in several of them related to learning analytics, educational data mining and adaptive systems. He is the co-author of more than 150 scientific publications including more than 50 in journals indexed in the JCR. In addition, he has coordinated the development and deployment of different learning analytics tools. He is also an IEEE Senior Member from 2015. His skills and experience include research and development in learning analytics, educational data mining, evaluation of learning experiences, user studies, gamification or Intelligent Tutoring System.

Carlos Delgado Kloos

Carlos Delgado Kloos received the Ph.D. degree in Computer Science from the Technische Universität München and in Telecommunications Engineering from the Universidad Politécnica de Madrid. He is Full Professor of Telematics Engineering and Rector's Delegate for Digital Microcredentials at Universidad Carlos III de Madrid, where he is also the Director of the GAST research group and Director of the UNESCO Chair on "Scalable Digital Education for All". He has carried out research stays at several universities such as Harvard, MIT, Munich, and Passau. His main research interests are in Educational Technology. He has been involved in a large number of research projects and has published around 500 articles. He has coordinated several MOOCs with over 600,000 registrations and is presently promoting the adoption of digital micro-credentials in Spain through the project CertiDigital (certidigital.es).