IADIS International Journal on WWW/Internet
Vol. 22, No. 2, pp. 16-29
ISSN: 1645-7641

A STRATEGY FOR PREDICTING STUDENT PERFORMANCE ON AN ONLINE PLATFORM: PROPOSAL, DEVELOPMENT AND VALIDATION

Gabriel Catizani Faria Oliveira and Guilherme Tavares de Assis
Department of Computing - UFOP, Ouro Preto - MG, Brazil

ABSTRACT

One of the potential applications of prediction models in education is in online learning. Considering online education, the gamified platform TôSabendo ("now I know" in portuguese) was created based on quizzes (question and answer games) with the aim of generating engaging experiences in Higher Education Institutions. The intention is to create a challenging environment for the player, motivating them to learn the concepts presented in each question and giving them a sense of progression in the task at hand. However, the platform currently lacks a prediction strategy using prediction models to help teachers understand, through predicted knowledge, how a particular student may perform on the platform. This understanding would be valuable for improving teaching methods and the content activities of classroom subjects, both in the traditional classroom setting and on the TôSabendo platform itself. Therefore, the goal was to propose, develop, and validate a strategy for predicting student performance on the TôSabendo platform. With the proposed and developed prediction strategy, a practical experimentation was conducted involving different prediction models and fictitious datasets. The evaluation initially assessed the model that would perform best with 10 different datasets, one for novice students and another for veterans. Subsequently, the models that achieved the best results in this first experiment go through an evaluation of different hyperparameters. Overall, after evaluating the models, decision trees yielded the most satisfactory results for both novices and veterans. By further refining this model through training with different hyperparameters, accuracy and precision results close to or equal to 100% were obtained, a value that must be analyzed and evaluated in the future due to the need to create synthetic data, which suggests a possible overfitting.

KEYWORDS

Prediction Strategy, Prediction Models, TôSabendo Platform, Student Performance

1. INTRODUCTION

Currently and frequently, game elements have been utilized in digital systems to offer students ways to engage more effectively in the learning process of the content discussed in the classroom. According to Prensky (2001), the new generation of students, known as "digital natives," has a natural affinity for technology and gaming experiences, making gamification an effective strategy to capture their attention and motivation. Games promote educational scenarios, "providing students with learning experiences that might not be as easily achieved through traditional teaching methods" (Giardinetto & Mariani, 2005).

Following this view of gamification in education, a digital platform called TôSabendo (“now I know” in portuguese), an extracurricular resource, was developed and validated, based on quizzes that aim to create a challenging environment for the player, motivating them to learn the concepts presented in each question and giving them a sense of progression in the task they are performing. Further details about the TôSabendo platform are described in Ferreira (2022) and França et al. (2021).

To enhance the TôSabendo platform based on the data generated by the students who use it, combining it with prediction models is an excellent alternative. According to Alyahyan and Düştegör (2020), prediction models can bring significant benefits to the learning process, such as personalized and targeted feedback to students based on the analysis of data generated by the digital environment. This way, immediate feedback can offer specific information about the areas in which a particular student is excelling or needs improvement. Thus, incorporating a prediction model into the TôSabendo platform, the main objective of this work, can help understand the academic difficulties and strengths of the students who use it, as well as predict the performance of future students registered on the platform.

The rest of this paper is organized as follows. Section 2 addresses related work. Section 3 presents the proposed strategy for predicting students' performance. Section 4 describes the experimental evaluation on the defined strategies and the results obtained. Finally, Section 5 concludes the paper and gives some directions for future work.

2. RELATED WORK

Several current studies are focused on predicting the performance of higher education students using various parameters and prediction models. Aiming to better organize the related works, Subsections 2.1 and 2.2 discuss, respectively, works on predicting semester performance and performance in online exams.

2.1 Prediction of Academic Performance

Yağcı (2022) experimented with various models for predicting student performance using classification algorithms and a dataset consisting of 1854 student records from a single course at a state university in Turkey. After data processing, selecting the best parameters and variables, and discretizing the final exam scores of these students, the Naive Bayes, Random Forest, KNN, ANN, and Logistic Regression models were trained and tested. Table 1 presents the results obtained considering the AUC, accuracy, F1-measure, precision, and recall metrics; it is observed that the ANN and Random Forest models performed the best.

Table 1. AUC, Accuracy, F1-measure, Precision, and Recall of the models (Yağcı, 2022)

Model	(AUC)	Classification accuracy (CA)	F1	Precision	Recall
Random Forest	0.860	0.746	0.721	0.752	0.746
Neural Network	0.863	0.746	0.723	0.748	0.746
SVM	0.804	0.735	0.704	0.735	0.735
Logistic Regression	0.826	0.717	0.685	0.700	0.717
Naïve Bayes	0.810	0.713	0.692	0.706	0.713
kNN	0.810	0.699	0.694	0.691	0.699

Unlike Yağcı (2022), Kumar & Vijayalakshmi (2011) evaluated student performance using a prediction strategy based on Decision Tree algorithms: J48 and ID3. For this, records of 115 computer science students from PRIST college were used—a smaller number of instances compared to the previous work but with well-chosen prediction criteria (grades from 5 subjects involving exams and assignments). The results were satisfactory for both J48 and ID3 models: accuracy of 92.2% and 88.8%, respectively. However, more instances need to be added for training and testing the model implemented by Decision Trees.

Therefore, these two articles have objectives similar to this work, as they present and test strategies with various prediction models to predict student academic performance. In contrast, our work focuses on predicting performance on an online academic platform. According to Han et al. (2022), these studies can help HEIs establish a learning analytics framework and contribute to decision-making processes.

2.2 Prediction of Performance in Online Exams

Tomasevic et al. (2020) conducted the prediction of student performance in online exams, which is particularly important for identifying students who are likely to fail the final exam so that additional assistance or tutoring can be provided in time. For this, Tomasevic et al. (2020) used a large publicly available dataset called the Open University Learning Analytics Dataset (OULAD) (Kuzilek et al., 2017).

The performance analysis consisted of factors such as student demographics, student performance in course assessments, and student engagement data. More precisely, student performance considered scores on intermediate assessments and the final course exam, and the number of exam attempts. This closely resembles how the TôSabendo platform functions: the quizzes, containing intermediate and a final quiz with all previously presented topics, can be taken as many times as necessary, with the final quiz being the most important and truly indicating if the student "knows it".

For classification, the algorithms KNN, SVM, Neural Networks, Decision Tree, Naive Bayes, and Logistic Regression were considered, using the F1 metric, and three types of data: demographics (D), engagement (E), and performance (P). Analyzing the obtained results, it was noticed that the highest effectiveness (highest F1 value) was achieved by exploring all three types of available data (D + E + P) in the KNN, SVM, Neural Networks, Decision Tree, and Logistic Regression models. As for the Naive Bayes and ANN models, the highest effectiveness was achieved by considering, respectively, the combinations (D + P) and (E + P). According to Garg (2018), whether or not to use demographic data along with these other engagement and

performance data has different opinions in the literature. If demographic data exists in the database, it is recommended to use it; otherwise, it is not of great importance, and there is no need to explore.

Therefore, it is observed that the most effective data types for general student performance prediction are performance and demographics. For prior student performance, the most commonly used data are assessment scores, exams, extracurricular activities, and overall GPA. For student demographics, the most commonly used data are gender and age. Although prior performance has shown to be the most impactful in predicting student performance, combining two or more types of data has been shown to further assist in prediction. Thus, these data types and prediction models, with their appropriate parameters, were tested to develop and evaluate the prediction strategy proposed in this work, focusing on predicting student performance on the TôSabendo platform.

3. PROPOSED PREDICTION STRATEGY

As previously mentioned, the general objective of this work is to propose, develop, and validate a strategy for predicting the performance of students on the TôSabendo platform, aiming to more comprehensively understand student performance when addressing the platform's quiz questions. To this end, several studies were conducted to predict student performance. Among them, various prediction models and different Educational Data Mining[1] (EDM) techniques were explored to achieve this goal.

Based on these studies, this Section aims to provide a detailed presentation of the proposed strategy and the rationale behind each decision made in its composition. Accordingly, the structure is outlined as follows: Section 3.1 presents the operational architectures associated with the proposed strategy to achieve the overall objective of this work, while Section 3.2 describes the database remodeling of TôSabendo to enable the integration of the proposed predictive strategy into the platform.

3.1 Working Architectures

To establish good prediction models for the proposed prediction strategy in this work, an architecture was initially defined to verify different prediction models following processes of EDM. This architecture is described in Figure 1. Generally, the flow starts with brief data processing, followed by applying and analyzing various prediction models. The students were divided into two groups: novice students, who have just entered the institution, and veteran students, who had completed at least one period. The steps of the architecture were performed separately for each group due to their different prediction attributes.

According to the architecture presented in Figure 1, Step 1 consists of a selection of performance factors, focusing on those considered most important for a given prediction. As described by Dutt et al. (2017), various factors can directly influence student performance, such as prior academic performance, student demographics, and the school environment. In this step,

[1]Educational Data Mining (EDM) is an emerging research field in its own right, concerned with developing methods to explore the increasingly large and unique data that comes from educational environments (Alyahyan & Düştegör, 2020; Liñán & Pérez, 2015; Dutt et al., 2017).

given that there are two groups of students, the factor selection will differ for each group. For example, novice students do not yet have academic grades, so only their pre-academic performance with school grades needs to be used, while veteran students have both school and academic grades. Next, in Step 2, synthetic data of platform students are provided, based on the selected relevant performance factors from Step 1, which are used to validate prediction models. After that, in Step 3, data preprocessing is done, which is a step prior to the implementation and application of prediction models. Here, the data undergo final processing before being used for training and testing the models (Osborne, 2002). Subsequently, in Step 4, a prediction model analysis is carried out, using the preprocessed data from Step 3 to apply, evaluate, and interpret them. Thus, the best prediction models to be added to the TôSabendo platform are identified, one for novice students and another for veteran students. Finally, in Step 5, the best prediction models are selected, one for novice students and another for veterans, to be used on the TôSabendo platform to predict performance whenever new students register for the first time.

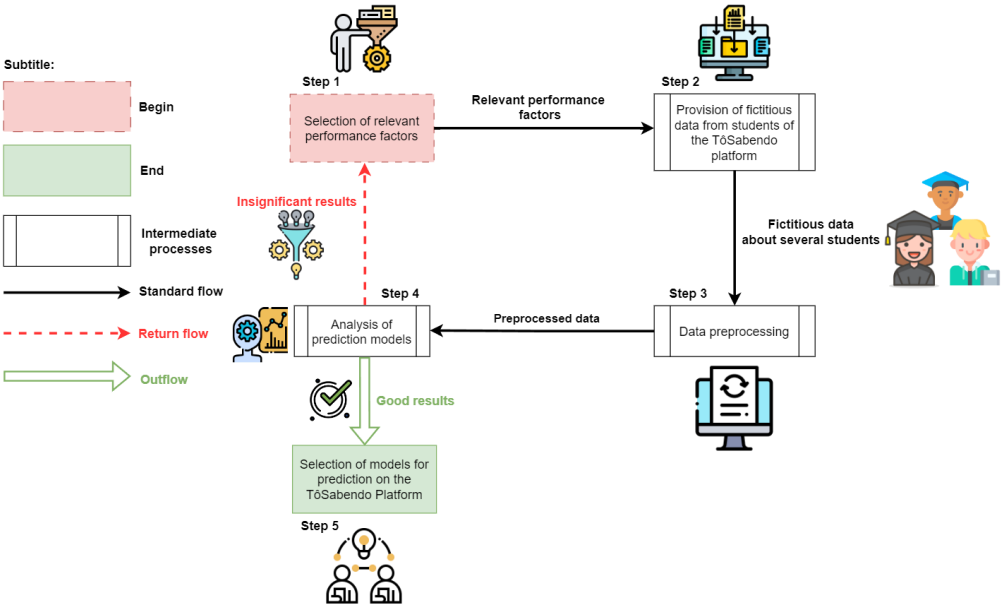

Figure 1. Verification of prediction models for the TôSabendo platform

Particularly, considering only Step 4 of Figure 1, an architecture (see Figure 2) was developed to demonstrate how it was performed. This architecture aimed to find, applied separately, the best prediction model for novice students and veterans. According to Ruano et al. (2010), there are various ways to find the best way to use these models to obtain the best information, knowing their parameters; however, the simplest and easiest is the trial-and-error approach, which involves conducting numerous experiments by modifying parameter values until the most beneficial performance parameters are found.

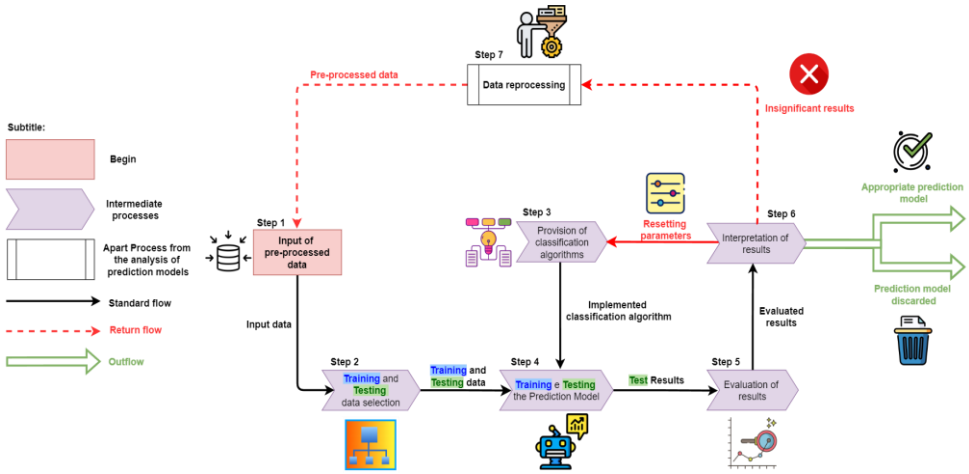

Figure 2. Prediction model analysis

According to the architecture presented in Figure 2, Step 1 consists of inputting data that have been fully processed and are used as input for the prediction model analysis. Next, in Step 2, the input data from Step 1 are divided, selecting which data are for training and which are for testing. This selection can be done in various ways; however, initially, the most straightforward method is used, separating the data into percentages, such as 70% for training and 30% for testing. Subsequently, in Step 3, a classification algorithm is provided to develop the desired prediction model. The classification algorithms used are Decision Trees, KNN, Naive Bayes, or Neural Networks. After that, in Step 4, the prediction model is applied to the input data selected for training and testing from Step 2 and with the implemented classification algorithm from Step 3. Next, in Step 5, the results of the prediction model's test and training from Step 4 are evaluated. For this, effectiveness measures such as accuracy and precision are applied, which are established through a confusion matrix (Nogare, 2020). This evaluation indicates, through comparisons, which data were correctly predicted, that is, if the student predicted to perform well actually did well on the platform or if the student predicted to perform poorly did not do well (Alyahyan & Düştegör, 2020). Then, in Step 6, the results evaluated in Step 5 are manually interpreted by visualizing graphs and tables with the metric values found. Through this interpretation, two return flows can arise: the first considers that the classification algorithm hyperparameters were poorly defined and need to be redefined, returning to Step 3. The second and most labor-intensive considers that the model training did not achieve good results in any of the occasions, even by altering the parameters; in this case, it is necessary to reprocess all the data (Step 7). As an output flow, after interpreting that the model's results were always worse than other models, even by redoing Step 3 and performing Step 7, the use of the model is discarded. However, if the results are significant and much better than other models, this prediction model will be appropriate for the TôSabendo platform, whether for novice students or veterans. Finally, in Step 7, data reprocessing involves Steps 1 to 3 of the architecture in Figure 1. Therefore, it is a new data treatment in the pursuit of improved results, where the subsequently processed data are used as input for the model (Step 1).

3.2 Database Remodeling of TôSabendo

For the proper handling of data from the TôSabendo Platform, it was essential to remodel the database to incorporate new information pertinent to the prediction of student performance, address existing inconsistencies, and ultimately deploy it within a database management system. Accordingly, Subsections 3.2.1 and 3.2.2 detail, respectively, the updates made to the conceptual Enhanced Entity-Relationship (EER) schema to include the relevant new information and the procedures followed for deploying the database within a management system.

3.2.1 Update of the Conceptual EER Schema

To update the conceptual EER schema, the TerraER[2] tool was employed to incorporate new data, such as student academic histories, including their grades, completed courses, and performance predictions. As illustrated in Figure 3, new entities, attributes, and relationships were added, with a primary focus on facilitating student performance prediction.

In addition to the academic history, the student's high school transcript was considered an important performance factor, especially for novices who do not yet have academic grades. Therefore, it was deemed necessary to include an entity called High School Transcript, which stores the average grade, the high school attended, the graduation date, and the average grades in the most relevant subjects related to higher education courses, namely Portuguese and Mathematics. It is important to emphasize that each student has only one academic history and one high school transcript. Thus, the attribute that uniquely identifies them is the student's "id" which corresponds to their CPF (Brazilian individual taxpayer registry number).

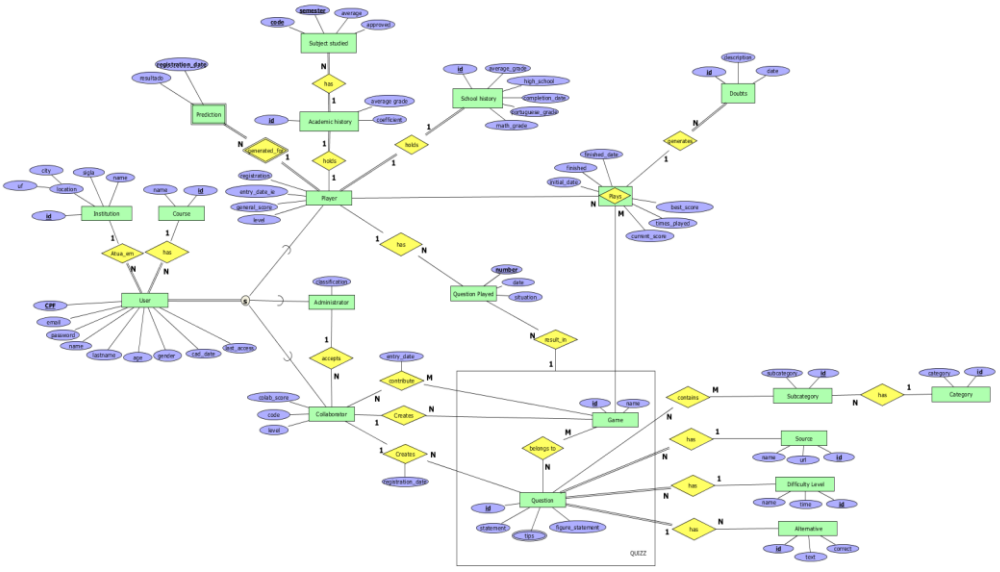

Figure 3. Updated Conceptual EER Schema

[2]TerraER is an open-source modeling tool designed to assist students in creating Entity-Relationship models. It is available at http://www.terraer.com.br/

Furthermore, a Prediction entity was added specifically to store the predictions made regarding student performance on the platform. In this regard, it was considered that a student may have multiple predictions, each uniquely identified by the student's "CPF" along with the date the prediction was made. The "result" attribute identifies the outcome of the prediction made by the predictive models, which can be "good performance," "average performance," or "poor performance".

Finally, some data and definitions in the database were modified to make it more consistent, including:

- the primary key of the User entity was changed from email to CPF, as it is also a unique attribute and easier to handle in database queries and writes;
- exclusion of the Location entity and replacement with a "location" attribute in the Institution entity, as there is no need to store the user's location, but rather the location of the higher education institution (IES) they attend;
- addition of the "age" attribute to the User entity, referring to the user's age, to be used as a demographic factor for student prediction;
- complementation of the Player (student) entity with the attributes "enrollment number" and "admission date to the IES," with the aim of obtaining their history and determining whether they are a freshman or a veteran;
- addition of a new Doubt entity, which indicates the students' doubts generated while they are playing a quiz, storing the description and the date it was raised.

Through the completed conceptual EER schema (Figure 3), the relational schema was modeled, which was then used for the database deployment.

3.2.2 Database Deployment

The new database was deployed in the PostgreSQL database management system following the conceptual EER schema. To do so, a tool called Prisma was used: an Object-Relational Mapping (ORM) for Typescript and Node.js programming languages, which were used to implement the platform's back-end.

An ORM facilitates and automates the integration and communication between relational database management systems and object-oriented programming languages (Kattah, 2024), such as Typescript. It plays a key role in data persistence, which is the ability to store and retrieve information consistently and efficiently in a database. The main advantage of an ORM like Prisma is that it allows for data model abstraction, meaning developers interact with data at a higher level of abstraction, treating database records as objects rather than rows in tables. Additionally, it increases productivity by reducing the amount of code required to manipulate data, as many CRUD-related tasks are automatically handled by the ORM.

For the creation of each table in the database using Prisma, a model was implemented, which defines all the attributes and their types, including primary keys and referential integrity constraints. After the models were created, a migration operation was performed, which transforms all these models into table creation queries and constraints, connecting to PostgreSQL and creating the tables in the desired database.

An example of a model representing the "User" table is shown in Figure 4, and its respective queries are shown in Figure 5.

```
model User {
    cpf              String   @id @db.VarChar(12)
    email            String   @unique @db.VarChar(100)
    password         String   @db.VarChar(100)
    firstName        String   @db.Text
    lastName         String   @db.Text
    gender           String   @db.Char(2)
    age              Int
    registrationDate DateTime @default(now())
    updatedAt        DateTime @updatedAt
    lastAccess       DateTime
    courseId         Int
    institutionId    Int

    course      Course      @relation(fields: [courseId], references: [id], onDelete: Restrict, onUpdate: Restrict)
    institution Institution @relation(fields: [institutionId], references: [id], onDelete: Restrict, onUpdate: Restrict)

    players        Player[]
    collaborators  Collaborator[]
    administrators Administrator[]

    @@map("users")
}
```

Figure 4. Model of the User table

```
-- CreateTable
CREATE TABLE "users" (
    "cpf" VARCHAR(12) NOT NULL,
    "email" VARCHAR(100) NOT NULL,
    "password" VARCHAR(100) NOT NULL,
    "first_name" TEXT NOT NULL,
    "last_name" TEXT NOT NULL,
    "gender" CHAR(2) NOT NULL,
    "age" INTEGER NOT NULL,
    "registration_date" TIMESTAMP(3) NOT NULL DEFAULT CURRENT_TIMESTAMP,
    "updated_at" TIMESTAMP(3) NOT NULL,
    "last_access" TIMESTAMP(3) NOT NULL,
    "course_id" INTEGER NOT NULL,
    "institution_id" INTEGER NOT NULL,

    CONSTRAINT "users_pkey" PRIMARY KEY ("cpf")
);

-- CreateIndex
CREATE UNIQUE INDEX "users_email_key" ON "users"("email");

-- AddForeignKey
ALTER TABLE "users" ADD CONSTRAINT "users_course_id_fkey" FOREIGN KEY ("course_id") REFERENCES "courses"("id") ON DELETE RESTRICT ON UPDATE RESTRICT;

-- AddForeignKey
ALTER TABLE "users" ADD CONSTRAINT "users_institution_id_fkey" FOREIGN KEY ("institution_id") REFERENCES "institutions"("id") ON DELETE RESTRICT ON UPDATE RESTRICT;
```

Figure 5. Queries for creating the User table

4. EXPERIMENTAL EVALUATION

In order to validate the proposed prediction strategy for student performance in TôSabendo on section 3, we performed an experimental evaluation described on Subsection 4.1; the obtained results are described and analyzed on Subsection 4.2.

4.1 Experimental Description

To evaluate the proposed prediction strategy, different datasets and prediction models (Decision Trees, KNN, Naive Bayes, and Neural Networks) were tested to analyze and interpret which datasets are best for novice and veteran students, respectively, and which are the best models

for these datasets, considering the architectures presented in Figures 1 and 2. Thus, two experiments were conducted: the first experiment (see Subsection 4.1.1) was to select which model had the best results for novice and veteran students separately; in the second experiment (see Subsection 4.1.2), these best models go through a selection of hyperparameters to further optimize their performance.

To analyze the experiments, accuracy and precision metrics were chosen, which together provide significant evaluation effectiveness. Accuracy is a simple and intuitive metric widely used in evaluating educational prediction models. Precision, on the other hand, focuses on the proportion of true positives relative to the total positive predictions of the model, identifying the model's ability to avoid false positives.

4.1.1 Determining the Best Models

Considering the architecture in Figure 1, to investigate the optimal performance of each model, regardless of hyperparameter selection, 20 distinct synthetic datasets were generated, consisting of 10 sets for novice students and 10 sets for veterans. Each set comprised 1000 instances representing different students. These datasets were generated synthetically and randomly rather than manually for several reasons that contribute to the effectiveness and robustness of the prediction models, namely: (a) encompassing a wide range of possible scenarios by considering various combinations of variables and values, ensuring that the models are trained and tested in diverse situations, making them more generalizable and capable of handling the natural variability of real data; (b) providing a more balanced and realistic representation of the possible situations the models may encounter, thus reducing the risk of overfitting where the model fits too closely to a particular dataset and may perform unsatisfactorily with new data.

In creating these synthetic datasets, the chosen attributes were prior academic performance and demographic data. For novice students, prior academic performance included average grades in Portuguese and Mathematics in high school, overall GPA, and the number of absences. For veteran students, prior academic performance was represented by their overall GPA and coefficient. In both categories, demographic data included the students' gender and age. Based on these characteristics, labels were assigned to each student, indicating whether their performance was classified as good, average, or poor.

After creating the student instances along with their respective labels, the data go through a preprocessing procedure to ensure quality and relevance in model learning. From the preprocessed data, a division was made, allocating 70% for the training set and 30% for the test set in each of the 20 generated datasets. This approach resulted in training and test sets that were completely random and distinct from each other. Moving on to model training, to reinforce validation and ensure a more robust evaluation, the K-fold cross-validation technique was incorporated. This technique allows for a comprehensive assessment of model performance, mitigating sensitivity to variations in training and validation data; it also contributes to selecting a more generalizable and reliable model, preventing overfitting. This technique was applied and evaluated on all datasets. Thus, after training and testing the models, a global average of each model's performance on the training and test sets was calculated. Based on the interpretation of these results, the best models were defined for the categories of novice and veteran students.

4.1.2 Hyperparameter Testing

After selecting the prediction models for novice and veteran students, considering the architecture in Figure 2, different hyperparameters were tested to optimize the chosen models and improve their results. For this task, a new dataset was created for novice students and

another for veterans with the same characteristics and size as those created for the experiment in Subsection 4.1.1. With these datasets, a Grid-search was performed along with K-fold cross-validation. After executing GridSearchCV, the function identifies and returns the model that presents the best combination of hyperparameters, based on the chosen evaluation metrics, which were accuracy and precision. This systematic and automated process ensures the selection of optimized models aligned with the established performance criteria.

Thus, for each model with the best-selected configurations, tests were conducted using isolated subsets (test subset) to evaluate the models' ability to handle new data. In situations where the results did not meet expectations, a new test was conducted with GridSearchCV, exploring different hyperparameter values that had not been previously considered in the initial experimentation. After the meticulous execution of GridSearchCV, repeated analysis and interpretation of the results, and the absence of significant improvements, the trained and tested models that provided the most satisfactory results were considered ideal for predicting student performance on the TôSabendo platform. This rigorous selection process ensures the use of robust models, adjusted to the nuances of the data, and provides reliability in the predictions.

4.2 Experiments Results

In the context of the first experiment, which encompasses the four implemented prediction models, the results of average accuracy and precision on the training and test sets are found in Tables 2 and 3 for novice and veteran students, respectively. The best results among all models are in green. It should be highlighted that in this experiment, hyperparameter selection was not the primary metric for identifying the optimal model.

Table 2. Average performance of models (%) for novice students

Model	Train Accuracy	Test Accuracy	Train Precision	Test Precision
Decision Tree	**98.65**	**99.03**	**98.68**	**99.04**
KNN	95.21	95.83	95.28	95.89
Naive Bayes	86.14	85.27	86.42	85.39
Neural Networks	93.64	94.17	93.85	94.33

Initially, the absence of overfitting was observed in all models since they were trained using cross-validation on the 10 datasets for both novice and veteran students. Thus, when evaluated on the test sets, the models maintained consistent performance, indicating robust generalization capability. This finding ensures that the models not only avoided overfitting to the training data but also effectively generalized to new datasets. Furthermore, given the effective generalization of the models, the Decision Tree stands out, showing the highest accuracy and precision in both training and test sets. This analysis underscores the importance of selecting models appropriate to the nature of the data and underlying patterns of the problem.

Table 3. Average performance of models (%) for veteran students

Model	Train Accuracy	Test Accuracy	Train Precision	Test Precision
Decision Tree	**99.70**	**99.77**	**99.71**	**99.77**
KNN	96.66	96.97	96.60	96.99
Naive Bayes	94.85	94.10	95.67	94.96
Neural Networks	97.35	97.87	97.35	97.97

Therefore, based on the presented results and analysis, it is concluded that predicting student performance, both for novice and veteran students, on the TôSabendo platform, shows greater effectiveness when employing simpler models, such as Decision Trees. The implementation of cross-validation, as a preventive measure against overfitting, demonstrated the robustness of the models, ensuring their ability to generalize to new datasets. The results of each Decision Tree were very important, although they fall outside the focus of this article. Hence, diagrams and explanations of the decision trees will not be shown.

With the Decision Tree defined as the main model for both novice and veteran students, the task remained to identify the ideal hyperparameters that could further enhance prediction results. For the models, the hyperparameters tested were criterion, splitter, max_depth, min_samples_split, min_samples_leaf[3]. After testing, the best configuration for novice and veteran students is presented in Tables 4 and 5, respectively.

Table 4. Best hyperparameters in decision tree for novices

Hyperparameter	Best Value
Criterion	entropy
Splitter	best
Max Depth	None
Min Samples Split	1
Min Samples Leaf	2

Table 5. Best hyperparameters in decision tree for veterans

Hyperparameter	Best Value
Criterion	gini
Splitter	best
Max Depth	None
Min Samples Split	1
Min Samples Leaf	10

[3]Hyperparameters available at https://scikit-learn.org/stable/modules/generated/ sklearn.tree.DecisionTreeClassifier.html

With the selection of these hyperparameters, the results regarding accuracy and precision on the training and test sets for novice and veteran students are found in Tables 6 and 7, respectively. The presented data indicate an exceptional performance for the classification models applied to the groups of novice and veteran students: the training and test sets demonstrated accuracy and precision of 100% or nearly so, suggesting a remarkable learning and generalization capability of the models. However, such results may raise concerns about the possibility of overfitting the training data, even with the implementation of K-fold cross-validation.

Table 6. Training and testing results (%) for novices

Metric	Train	Test
Accuracy	100.0	99.33
Precision	100.0	99.33

Table 7. Training and testing results (%) for veterans

Metric	Train	Test
Accuracy	100.0	100
Precision	100.0	100

5. CONCLUSION

Given the findings in Section 4, it is crucial to consider that the performance of the models may vary depending on the nature of the data, the sample size, and the specific characteristics of the students being predicted, especially when considering synthetic data. In this context, it becomes necessary to conduct future tests to assess whether the Decision Tree will remain the most accurate choice for both categories of students. The use of real student data from the platform users could potentially favor Neural Networks, which are capable of identifying more complex data patterns, resulting in superior performance. However, given the initial experimental results, Decision Trees should be employed for predicting student performance.

As future work, we intend to: (1) create an automatic retraining system for the prediction models based on new student data; (2) conduct new experiments considering hyperparameter testing in the models to fine-tune them before selecting the best one; (3) experiment with how the data will perform with models that use regression algorithms to predict the scores students will achieve on the platform; (4) apply statistical significance tests to the presented results to check for overfitting caused by the creation of synthetic data.

ACKNOWLEDGEMENT

This research was partially funded by research grants from PIP/UFOP. Furthermore, it was carried out on the GAID/UFOP Laboratory.

REFERENCES

Alyahyan, E. and Düştegör, D., 2020. Predicting academic success in higher education: literature review and best practices. *International Journal of Educational Technology in Higher Education*, Vol. 17, No. 1, 3.

Dutt, A., Ismail, M. A. and Herawan, T., 2017. A systematic review on educational data mining. *IEEE Access*, Vol. 5, pp. 15991-16005.

Ferreira, C. O., 2022. Desenvolvimento de uma estratégia de machine learning para aprimoramento da plataforma Tôsabendo. In *UFOP*. Ouro Preto, MG.

França, T. F. et al., 2021. Tôsabendo: A platform to create engaging teaching and learning experiences. *2021 XVI Latin American Conference on Learning Technologies* (LACLO). IEEE, pp. 275-281.

Garg, R., 2018. Predicting student performance of different regions of Punjab using classification techniques. *International Journal of Advanced Research in Computer Science*, Vol. 9, No. 1, pp. 236-241.

Giardinetto, J. R. B. and Mariani, J. M., 2005. Jogos, brinquedos e brincadeiras: O processo ensino-aprendizagem da matemática na educação infantil. *Matemática e Educação Infantil*. Universidade Estadual Paulista "Júlio Mesquita Filho".

Han, J., Pei, J. and Tong, H., 2022. *Data mining: concepts and techniques*. Morgan Kaufmann.

Kattah, A., 2024. O que é ORM – entenda a importância e como utlizar na programação. *Hero Code*. Available at: https://herocode.com.br/blog/o-que-e-orm/ (Accessed: 17 November 2024).

Kumar, S. and Vijayalakshmi, M. N., 2011. Efficiency of decision trees in predicting student's academic performance. *Computer Science & Information Technology*, Vol. 2, pp. 335-343. https://doi.org/10.5121/csit.2011.1230

Kuzilek, J., Hlosta, M. and Zdrahal, Z., 2017. Open university learning analytics dataset. *Scientific Data*, Vol. 4, No. 1, pp. 1-8.

Liñán, L. C. and Pérez, Á. A. J., 2015. Educational Data Mining and Learning Analytics: differences, similarities, and time evolution. *Universities and Knowledge Society Journal*, Vol. 12, No. 3, pp. 98-112.

Nogare, D., 2020. Performance de Machine Learning – Matriz de Confusão. *Diego Nogare: Inteligência Artificial & Machine Learning*. Available at: https://diegonogare.net/2020/04/performance-de-machine-learning-matriz-de-confusao (Accessed: 11 July 2023).

Osborne, J., 2002. Notes on the use of data transformations. *Practical Assessment, Research, and Evaluation*, Vol. 8, No. 1.

Prensky, M., 2001. Digital natives, digital immigrants, Part II: Do they really think differently? *On the Horizon*, Vol. 9, No. 6, pp. 1-6.

Ruano, M. V., Ribes, J., Sin, G., Seco, A. and Ferrer, J., 2010. A systematic approach for fine-tuning of fuzzy controllers applied to WWTPs. *Environmental Modelling & Software*, Vol. 25, No. 5, pp. 670-676.

Tomasevic, N., Gvozdenovic, N. and Vranes, S., 2020. An overview and comparison of supervised data mining techniques for student exam performance prediction. *Computers & education*, Vol. 143, 103676.

Yağcı, M., 2022. Educational data mining: prediction of students' academic performance using machine learning algorithms. *Smart Learning Environments*, Vol. 9, No. 1, 11.