Student's Academic Performance Prediction in The Programming Techniques Course By Using Machine Learning Methods

Quoc Huy-Le
Faculty of Information Technology
Industrial University of Ho Chi Minh City
22636191.huy@student.iuh.edu.vn

Thiet Pham-Thi*
Faculty of Information Technology
Industrial University of Ho Chi Minh City
*phamthithiet@iuh.edu.vn

Nguyen Huynh-Tuong
Faculty of Information Technology
Industrial University of Ho Chi Minh City
htnguyen@iuh.edu.vn

Khoa Vo
Faculty of Information Technology
Industrial University of Ho Chi Minh City
khoavo@iuh.edu.vn

Tien Vu-Van
Faculty of Computer Science and Engineering
Ho Chi Minh City, Vietnam
vvtien@hcmut.edu.vn

Huy Tran
Faculty of Computer Science and Engineering
Ho Chi Minh City, Vietnam
tranhuy@hcmut.edu.vn

***Abstract*— Learning Management System (LMS) plays an essential role in interactive learning, especially after the COVID-19 outbreak with a rapid shift from traditional classroom learning to online learning. LMS also facilitates the continuous collection and evaluation of students' performance. In this paper, we focus on predicting students' in-class performance (i.e., in-lab) from their homework performance (i.e., pre-lab) using the data collected from LMS with four experimental exercises for the programming techniques course. We also investigate several machine learning methods and hyperparameters for model selection. The results show that Random Forest has the best performance. This investigated cause-effect relationship between pre- and in-lab grades not only helps teachers with a better understanding of the curriculum but also students with their learning results.**

Keywords*— *Machine learning models, Programming techniques course, Student' academic performance.

I. INTRODUCTION

Recent years, the rise of technology and digital platforms has ignited a transformation in education, reshaping teaching methods and opening doors to the vast potential of online learning. The rapid increase of online learning is especially remarkable, a trend that has been further highlighted in the unique global pandemic situation triggered by COVID-19. The outbreak necessitated a swift shift to remote learning solutions, highlighting the essential role of Learning Management Systems (LMS) in ensuring uninterrupted teaching.

The LMS platform serves as a digital framework facilitating interaction between learners and educational content, additionally playing a crucial role in gathering valuable learner data, paving the way for profound insights based on data regarding both the learning process and the learners themselves, this abundant source of data holds immense potential in enhancing our understanding of educational programs and student behavior, ultimately contributing to the creation of more effective teaching and learning methods.

For the course of Programming Techniques, learning requires a lot of patience and determination, indicating the need for extensive practice at home to improve programming skills. Self-study exercises offer students a highly efficient method to actively participate in practical hands-on practice. Moreoover, programming questions and answers have their own characteristics, different from conventional questions. To check the correctness of their programming assignments, students can submit their tasks multiple times on testing platforms through the LMS learning system. The results are then returned immediately, enabling students to debug their code and improve it. This environment creates many self-learning opportunities for students.

Furthermore, the dynamic nature of the LMS platform allows for timely intervention and support for learners by identifying important issues and providing relevant feedback. By utilizing the power of machine learning algorithms, we are able to effectively utilize this data to create advanced predictive models that have the capability to analyze and evaluate the learning paths of students, ultimately providing timely notifications to aid and enhance their academic advancement.

In recent years, leveraging educational data to assess and predict students' academic performance has been a growing field of research aimed at enhancing students' learning outcomes. Algorithms such as Logistic Regression and Naive Bayes have been widely applied in practical scenarios such as stock market prediction, medical data forecasting, and educational data analysis. Besides, emotions and attention in the learning environment are also imperative factors that induce good academic achievement in any educational sector. With the purpose of assessing learners' attention during the lesson, Angeline et al. [1] have proposed an approach to identify the learners' emotions in the learning environment by using machine learning algorithms. This approach implemented throught 2 stages. The first stage is to identify the facial emotions that are performed by using naïve bayes algorithm. The second stage, based on the emotional records, the academic achievement is analyzed with random forest algorithm. Their research results showed that the attention of the learners in the learning environment with positive emotions have better academic results.

In [2][3][4][5] the experiments conducted have demonstrated fairly accurate predictions, providing valuable support for subsequent decision making processes. Teaching and learning programming subjects pose significant challenges for educational institutions involved in this domain. The ability to predict and anticipate students' learning outcomes can bring numerous benefits to educational institutions, students, and administrators, facilitating better learning and teaching practices. Specifically, it can enhance students' programming interaction skills and problem solving abilities.

By bridging the gap between data analysis and educational practice, our research endeavors to contribute to the ongoing discourse surrounding the optimization of the online learning experience. We believe that leveraging the power of machine learning combined with LMS platforms can pave the way for new avenues in personalized learning and foster continuous improvement in students' academic outcomes. Through this exploration, we aim to pave the way for an educational landscape that adapts to current development trends, meeting the demands for learner quality in the digital age.

At present, the trend in studying technical fields, especially in IT, often involves laboratory work both in class and at home. Students engage in learning and researching in the classroom, then supplement their studies at home after class. In the Programming Techniques course, we design four prelab assignments and four inlab assignments for students. It is mandatory for students to complete the prelab assignments before attending class. During class, the lecturer provides detailed guidance on completing the inlab assignments. The knowledge gained from the prelab assignments supports the inlab tasks. Based on the data collected from students' Learning Management System (LMS), we develop four corresponding datasets for the four lab assignments.

As mentioned earlier, students are required to complete Prelab exercises at home before attending class, where instructors guide them through the Inlab exercises. Throughout the learning process, subsequent Inlab exercises will build upon the Prelab and previous Inlab exercises. We have utilized this to integrate into our machine learning model in order to closely track students' learning progress. Our research addresses the following questions:

- Is it possible to predict which students are struggling in a programming course so that instructors can provide them with support?
- What kind of history data can give in order to achieve the best prediction?

The following content of the article is distributed as follows: Section II of the article presents related works. The proposed approach of the article is presented in section III. Section IV is to present the experimental results and evaluate the model. Finally, conclusions and proposed development directions are presented in section V.

II. RELATED WORKS

Classification problem the process of using machine learning techniques to define features, classes based on different data and classify new instances based on unknown data. Features are used as input data, a model is trained using previously labeled examples, and a binary or multiclass class is predicted [6]. Some the classification algorithms employed in predicting student performance including Naive Bayes (NB), Support Vector for Classification (SVC), Decision Trees (DT), K Neighbors Classifier (KNN), Multilayer Perceptron (MLP), random forest (RF), gradient boosting classifier (GBC). Beside, Regression algorithms have used in the field of student performance prediction as Linear regression (LR), and logistic regression.

In the field of machine learning and predicting student learning outcomes, numerous studies and applications have been conducted. Previous research has focused on using machine learning methods to analyze learning data and predict student learning outcomes. Predicting students' inclass performance helps educators identify groups of students with low performance and guide them to improve their learning outcomes.

Some studies have utilized classical machine learning models such as linear regression, logistic regression, and support vector machines to predict students' academic grades based on input variables such as previous grades, class attendance, activities on online learning platforms, and many other factors.

Other studies have explored the use of data mining techniques to understand more about the student learning process. These methods often include data pattern analysis, time series analysis, and discovering hidden patterns in learning data.

Additionally, there are studies focusing on developing alert systems to provide feedback and support for students during their learning process. These systems often use machine learning algorithms to detect abnormal patterns in students' learning behavior and provide appropriate suggestions or alerts [7][8]. In order to mitigate the dropout rate among students in educational institutions, Uyen and colleagues [9] employed two models, Logistic Regression and Naive Bayes, on a dataset comprising entrance exam scores, scores from initial semesters, and current academic status to predict the likelihood of student dropout. Their research findings could assist institutions in issuing early warnings and reducing dropout rates for subsequent cohorts of students.

However, despite numerous studies on predicting student learning outcomes, applying these methods on online learning platforms still faces many challenges. These challenges include the complexity of online learning data, uncertainty in data collection, and the ability to integrate prediction systems into the teaching process and effectively support students.

In study [10], the authors apply machine learning models to predict students' academic grades based on online learning data. They use methods such as linear regression, support vector machines, and decision trees to create predictive models.

In study [11] focuses on predicting students' learning performance using supervised machine learning methods. Data is collected from B.I.S.E Peshawar, Khyber Pakhtunkhwa. This work has use the Regression and decision tree classifiers to predict students' grades and ranks based on their previous academic records.

[12] This study focuses on improving personalized learning outcomes for students by using dynamic testing accompanied by a prediction model. Specifically, the main goal is to create adaptive dynamic tests to assess students' learning performance while continuously comparing the evaluation results with the prediction model based on decision trees. This research provides deep insights into the effectiveness of this method in improving adaptive dynamic learning interactions.

The authors in study [13] have focused on predicting student dropout in online courses. They use decision tree method to classify students into different dropout risk groups based on previous learning and behavioral data.

In 2022, Thien-Wan and colleagues [14] utilized four classification models in the field of machine learning, namely Naive Bayes classifier, C4.5 Decision Tree classifier, KNN classifier, and Random Forest classifier, to predict the performance of programming students. In this study, the authors utilized the Weka Explorer environment to construct and evaluate the four classification models on datasets collected during practical programming courses. Based on experimental results, the authors concluded that the Random Forest Classifier model performed the best on the programming coursework dataset they gathered.

To enhance students' learning outcomes in programming courses, Jose Llanos and colleagues [15] in 2023 proposed a predictive model for students' performance in weeks 3, 5, and 7 out of the 16 weeks of a programming course. The authors employed eight classification algorithms including Naive Bayes (NB), Support Vector for Classification (SVC), Decision Trees (DT), K Neighbors Classifier (KNN), Multilayer Perceptron (MLP), Random Forest (RF), Gradient Boosting Classifier (GBC), and Linear Regression (LR) to train and evaluate the model on a dataset comprising three main features: scores, time spent, and the number of attempts by students on programming assignments and tests. To assess the performance of the models, the authors utilized metrics such as accuracy, recall, F1 score, and AUC. Based on experimental results, the authors suggested the use of the GBC model for effectively predicting students' learning outcomes in programming courses.

III. PROPOSED APPROACH

Our research focuses on evaluating students in the Programming techniques course. Because programming is a learning field that requires repetitive practice, making mistakes is inevitable. Preparing assignments before class will help students learn more. Our methodology of research is as shown in Figure 1:

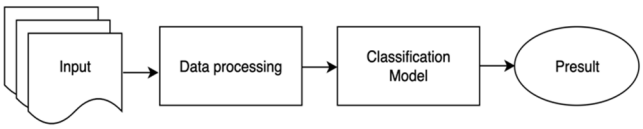

Figure 1: Methodology of research

Input in the methodology of research is datasets related to the results of programming techniques course. It has been practically implemented at an university including two semesters is that SEM-212 (acdemic year 2021-2022) and SEM-222 (acdemic year 2022-2023). This course organized 8 practical tasks divided into two types: 4 tasks conducted directly in class and 4 self-study tasks at home. By synthesizing all these practical tasks, we have created a complete dataset for analysis and research. The relationship between Prelab and Inlab is reciprocal. Prelab (Before class): Prelab can provide information about the preparedness of students before participating in class. Students who actively participate in Prelab often have a deep understanding of the course content and are better prepared for activities in Inlab. Data from Prelab, such as completing assignments, participating in pre-class discussions, or accessing course materials, can be used as indicators to measure student preparedness. Inlab (In class): Students' performance in Inlab activities can reflect their understanding and ability to apply knowledge in real-life situations. Results from exercises, group discussions, or tests in Inlab can be used to assess students' competence and academic performance. Active participation and contribution in Inlab can reflect students' interest and commitment to learning in class.

We have built a tool to extract student data from the LMS system for analysis. The results of extracting data from the system yielded 2 datasets:

1) new repost.csv: This dataset contains information about: Full Name, Student ID Number, Task Status, Start Time, End Time, Time Taken, Grade/10, Number of Questions Answered (out of 10), Question ID.

2) Question bank.csv: This dataset contains information about: Lab Type (Pre, In, Post), Question Name, Topic, Level, Question Link.

We will link these two datasets together through the Question IDs. Although our question dataset does not have the Question ID attribute, the attribute values (bkel-link) contain the Question IDs, allowing us to link these two datasets together. Then, we will generate a new attribute, tentatively called "attempt" (number of attempts), by counting the number of attempts made by each student for each lab task. We will split the "Lab" attribute into separate attributes. Since the "Lab" attribute has 4 values corresponding to 4 lab tasks, we will separate them for easier analysis. After completing the aforementioned steps, we will have a complete dataset consisting of 4 Prelab tasks and 4 Inlab tasks for each student. The structure of each Prelab and Inlab task will have the following corresponding features in table 1:

Table I: Description of experimental data

Feature	Description
Score	Student scores on Prelab/Inlab tasks comprise multiple questions, with each question allowing students to submit multiple times and the highest score is finally chosen.
Growths	This measure is based on changes in the number of correct answers, an increase in the difficulty of questions, or improvement in completion time.
Number of attempts	This is the number of attempts made by a student to complete a part of the lab or assignment before submitting the final version.
Number of questions	The number of questions solved by students can indicate the student's level of understanding of the content as well as their level of interaction and curiosity about the topic.
Time spent	The time spent completing a part of the lab or assignment can be an indicator of the student's effort level as well as the difficulty of the task.
Last submission	The last submission time can provide information about the student's time management skills as well as their preparedness and task management skills.

However, some students may not exert effort to complete assignments, so predicting their performance based on assignment data may not be feasible. Therefore, students with low attempt counts will be excluded from the dataset. To encourage carefulness in task completion, we have limited the number of attempts for each question to 5. For the number of questions in each practice task, denoted as "question count," the maximum attempts allowed is 5 as well. Additionally, we use the "Prelab-attempts" index to measure the number of attempts by students in each practice task. This is necessary because the number of attempts does not fully reflect the number of questions completed by students. For example, if a student completes 5 questions with each question attempted twice, the "Prelab-attempts" will be 10. Similarly, if a student completes 10 questions with each question attempted once, the "Prelab-attempts" will also be 10. This indicates that the number of attempts does not accurately reflect the number of questions completed in the practice task. Therefore, this factor is used for evaluation. In each practice task, this can be done by comparing the number of attempts (Prelab-attempts) with the maximum attempts allowed, which is 5, determined by the "question count." If a student's attempts for one practice task (whether Prelab or Inlab) exceed 5, this may indicate that the

student has completed many questions in the practice task. However, if the number of attempts is very low, it may indicate that the student has only partially completed the questions or has not completed the practice task.

After careful data collection and processing, we selected suitable classification models for the problem. These models include Random Forest, Logistic Regression, GaussianNB, and KNN. Next, we will proceed with the training process to build these models and evaluate their performance based on evaluation metrics such as accuracy and F1-score, precision, recall. This helps us select and fine-tune the best model to predict and classify student data accurately and effectively in the online learning environment. The result of this article is hoped to predict student grades from their previous generated data. We will attempt to build a machine learning model to predict student grades before they attend class in order to provide timely reminders to improve their grades and academic quality.

IV. EXPERIMENTAL RESULTS

A. Experimental Data

Experimental data includes 4 inlab and 4 prelab assignments for students as mentioned the above. The results of inlab assignments are extracted to predict the results of subsequent inlab assignments (each time a student attends class, they will receive a prediction from us) based on the prelab data of self-study students at home or by synthesizing prelab and previously learned inlab assignments to predict the nearest inlab assignment for students. The results of inlab assignments will be converted into labels according to the following rules:

- Score (>=0 and) < 4: Poor performance, label 0
- Score >= 4 and < 5.5: Average performance, label 1
- Score >= 5.5 and < 7.0: Good performance, label 2
- Score >= 7.0 and < 8.5: Excellent performance, label 3
- Score >= 8.5 (and <= 10): Outstanding performance, label 4

In this article, we embark on an effort to build a model to evaluate students' learning trajectories. Our model will utilize students' learning data to predict

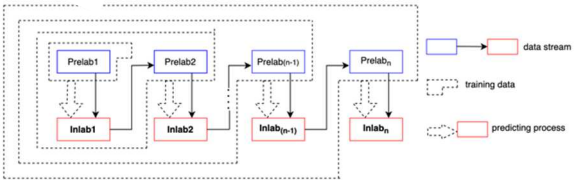

Figure 2: Prediction dataset overview

their in-class assignments, or in other words, we will use what the students have learned as training data. If a student has completed all four Inlab assignments, we will use four models to predict those four assignments, but the size of the training data will vary according to the students' learning progress, as follows:

- Prelab1 is used to predict Inlab1,
- Prelab1, Inlab1, and Prelab2 are used to predict Inlab2,
- Prelab1, Inlab1, Prelab2, Inlab2, and Prelab3 are used to predict Inlab3,
- Prelab1, Inlab1, Prelab2, Inlab2, Prelab3, Inlab3, and Prelab4 are used to predict Inlab4.

We chose this score range as the criterion for predicting student grades, in line with the current assessment methods in the education system of Vietnam.

Our task is to build a machine learning model to evaluate students' learning processes at home and at school according to the grading scale mentioned above, and the system will provide timely alerts to support students in improving their grades. This model will analyze data on students' learning behaviors, including studying at home and interacting with instructors during the Programming Technique course. The ultimate goal is to predict students' learning progress and provide specific suggestions for them to improve their academic performance.

In this study, we selected models such as Random Forest, Logistic Regression, MLP, and KNN to address the problem of evaluating students' learning processes and making predictions about future academic outcomes. Then, we will propose a model with the highest accuracy to apply in practice.

To evaluate the performance of the models, we used the Cross-validation technique. This is a common method in machine learning, which involves dividing the dataset into smaller parts and performing training and testing on these parts. In this case, we divided the dataset into 5 parts, where one part was reserved for testing and the remaining 4 parts were used for training. This process was repeated 5 times, each time using a different part as the test data and the rest as the train data.

In addition, to evaluate the performance of the proposed model in this study, the following metrics were used: accuracy, Precision, Recall, and F1-score, with specific calculation methods as follows: Recall: Recall is known as the rate of True Positive. It relates to the rate of correctly identified positive instances of all actual positive instances.

$$Recall = TP/(TP + FN) \quad (1)$$

Precision: Precision is the correctly classified positive rate. It is the number of instances that are predicted correctly:

$$Precision=TP/(TP+FP) \quad (2)$$

F1 − score : F1 − score is known as the harmonic mean of recall and precision.

$$F_1 = 2* (Recall * Precision)/(Recall + Precision) \quad (3)$$

Where TP is True Positives and FP is False Positives. Accuracy: Accuracy is a metric to measure the correctly predicts outcome. It is the number of correct predicted instances in by the total number of prediction instances.

After each round of Cross-validation, we recorded the accuracy of the model on the test set and calculated the average of these results. This allows us to evaluate the average performance of the model on different datasets. This helps us gain an overall understanding of the accuracy and stability of the Random Forest model in predicting students' academic outcomes.

Using the Cross-validation technique not only improves the feasibility and accuracy of the model but also helps assess the generalization of the model on new data. This is important to ensure that the model can generalize well on real-world data and avoid overfitting.

B. *Experimental Evaluation*

During the research process, we conducted experiments and compared various machine learning models to address this problem. These models include a range of methods from basic

machine learning to more complex models. However, to ensure the accuracy and performance of the model are optimized, we selected a final model from the number of models tested. This final model was chosen based on multiple factors, including prediction performance, stability of results, as well as the ability to effectively handle large datasets. Our results indicate that the model performs quite well.

We tried running several machine learning models such as Random Forest, Logistic Regression, NaiveBayes, K-nearest neighbors (KNN). Tables I above show the performance of the models when run with our dataset.

The table II above illustrates the performance of the models when applied to the experimental dataset with features extracted from 4 Inlab assignments and 4 Prelab assignments. The instability of NaiveBayes on Inlab3: NaiveBayes yields lower and unstable results on Inlab3, with notably lower Precision and F1-score compared to other models. This may indicate that the data distribution in Inlab3 does not align well with the assumption of independence between variables of NaiveBayes. Both Random Forest and KNN models, especially Random Forest, tend to perform the best on Inlab3 and Inlab4. This suggests that these datasets possess characteristics that decision tree-based models (such as Random Forest) or k-nearest neighbors (KNN) can effectively exploit. While both Random Forest and KNN perform well on Inlab2, Random Forest achieves higher Precision and F1-score than KNN.

Table II: Performance obtained for each predicted model

Prediction	Model	Precision	Recall	F1-score
Inlab1	Random Forest	0.845	0.831	0.785
	Logistic Regression	0.616	0.785	0.690
	NaiveBayes	0.616	0.785	0.690
	KNN	0.639	0.706	0.670
Inlab2	Random Forest	0.865	0.915	0.883
	Logistic Regression	0.813	0.901	0.855
	NaiveBayes	0.859	0.591	0.692
	KNN	0.829	0.886	0.855
Inlab3	Random Forest	0.905	0.928	0.897
	Logistic Regression	0.852	0.923	0.886
	NaiveBayes	0.918	0.461	0.592
	KNN	0.828	0.895	0.855
Inlab4	Random Forest	0.887	0.916	0.855
	Logistic Regression	0.816	0.903	0.857
	NaiveBayes	0.898	0.560	0.671
	KNN	0.837	0.898	0.860

This might suggest that Inlab2 has structures or features that Random Forest can learn and utilize better than KNN. Logistic Regression usually exhibits similar values of Precision, Recall, and F1-score, indicating a degree of stability in performance. However, these metrics are considerably lower compared to Random Forest, suggesting inferior real-world performance compared to Random Forest when deployed.

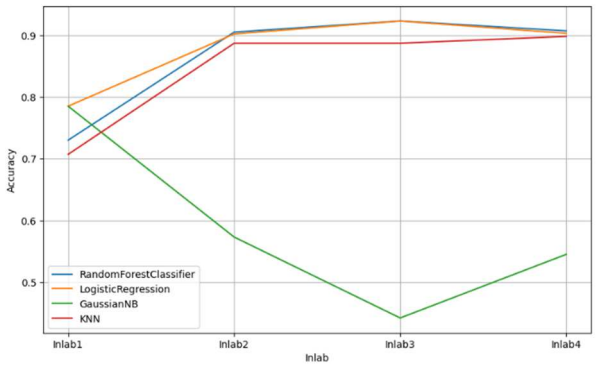

Figure 3: Accuracy of predicting models for four inlabs

With reference to Figure 3, it can be observed that, Random Forest Classifier achieved the highest performance in lab 3 with 0.928 and lab 2 with 0.9050. This means that this model works well on the data of the second and the third labs. However, the results in the first lab and the forth lab also achieve high performance, 0.730 and 0.9067 respectively. For Logistic Regression, the performance of this model is quite stable across labs, with results relatively close to each other. This model achieved the best results in the third and the second labs, similar to Random Forest Classifier. This implies that Logistic Regression is a good and stable choice for these datasets. The performance of the GaussianNB model is lower than other models, especially in the second and the third labs. However, in the first and the forth labs, the results are not too bad but still lower than other models. Similar to Random Forest Classifier and Logistic Regression, KNN achieves the best results in the third and the second labs. The results in the first lab are not bad, but the forth lab has the results with a bit lower than the other models. From the results of the Figure 3, we can see that Random Forest Classifier and Logistic Regression are the two best options based on the evaluation results in the labs, however, combining the experimental results in the Table II, the measuring results from Logistic Regression are not really high compared to the ones from Random Forest Classifier. For GaussianNB does not seem to be suitable for these datasets, while KNN also works well but not consistently across labs.

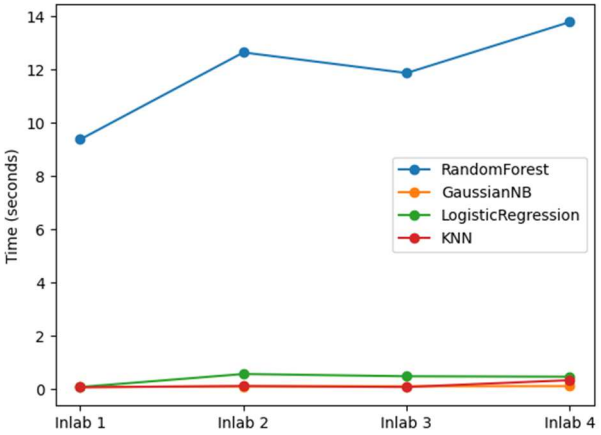

Figure 4: Training time of each model classifier

From Figure 4, it can be observed the training time of machine learning models (Random Forest, GaussianNB, Logistic Regression, and KNN) across training iterations from Inlab 1 to Inlab 4. Random Forest, although having relatively high training time compared to other models, shows a noticeable increasing trend with the number of training

iterations. Meanwhile, GaussianNB and Logistic Regression typically have low training times and exhibit little variation as the number of training iterations increases. KNN initially has relatively low training time but increases significantly in the final training iteration. This demonstrates the stability of the models and the impact of the number of training iterations on training time.

Based on the results from the Table II, the Figure 3, and the Figure 4, we propose using the Random Forest model for our classification problem. Although Random Forest has longer training time compared to other models, it provides us with stability in performance. The metrics of Accuracy, F1-score, Recall, and Precision show relatively small deviations from each other during training compared to other models.

We propose using the Random Forest model for practical applications because it is a machine learning method based on the principles of Ensemble Learning, where multiple models are combined to make the final prediction. A notable feature of Random Forest is the combination of multiple decision trees and randomness in selecting features and sample data to build these decision trees. The process of building a Random Forest model begins by generating a set of decision trees. Each decision tree is constructed by selecting a random subset of features from the dataset and a random subset of samples from the training data. This helps each decision tree to be trained on a small and diverse portion of the data. After creating the decision trees, the Random Forest model combines the results from all trees to make the final decision.

In this study, we found that the features related to time, "Time spent" and "Last submission", in both Prelab and Inlab, did not yield the expected effectiveness when used in the classification model. In fact, the information about time often does not accurately reflect the level of effort and diligence of students during the assignment completion process. Many students either open the assignment without working on it or leave their work idle for extended periods, creating noise in the training data. However, experimental results across models showed that time-related features such as "Time Spent" and "Last Submission" from Inlab and Prelab introduce noise into the training data and affect the classification performance of students. In reality, this information often does not accurately reflect the level of effort and diligence of students during assignment completion because many students open the assignment but do not engage seriously with it. They often leave the assignment open for hours without actively working on it, rendering time-related features ineffective for evaluating student performance.

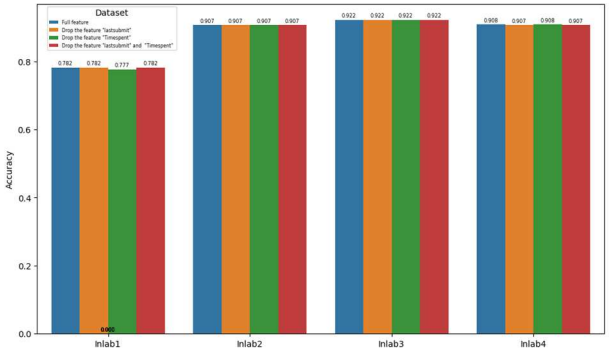

Figure 5: Accuracy of predicting models on the different features of dataset

Based on Figure 5, it can be seen that even after removing 1 or even 2 features, the Accuracy score remains unchanged. Therefore, we can confidently conclude that removing two features does not affect the model's performance. Furthermore, based on the observation that when removing 1 or both features, the training time of the model as shown in Figure 6 decreases significantly, it can be inferred that removing these two features is a sensible decision. This contributes to optimizing the Random Forest model, as one of its major drawbacks is its long training time, as seen in Figure 4, where its training time is much longer compared to other models. Although removing these two features does not affect the accuracy, it helps improve the model by reducing the training time. Hence, we have excluded the features "Time spent" and "Last submission" from our training model.

Based on the experimental results, the Random Forest model not only achieves stable performance but also provides the most accurate predictions compared to other models. Its flexibility and efficiency in handling large datasets have highlighted its superiority over other models. This demonstrates the suitability of the Random Forest model in addressing the evaluation and prediction of students' learning processes.

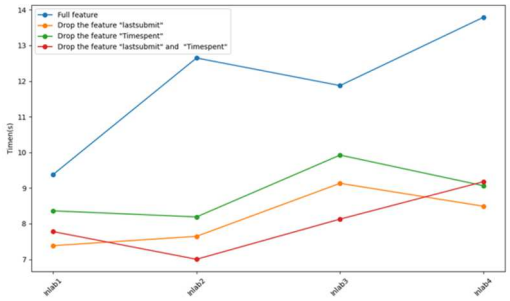

Figure 6: Training time of Random Forest Classifier on the different features of dataset

Based on the experimental results in this section, the Random Forest model has achieved the highest accuracy in all training iterations, specifically: Inlab1: 0.730, Inlab2: 0.905, Inlab3: 0.928, Inlab4: 0.905. Therefore, the paper proposes using the Random Forest model as the best method for evaluating and predicting students' learning processes.

V. CONCLUSION AND FUTURE WORKS

In this article, we have proposed a method to predict the practical learning outcomes of programming courses based on student's preparation process. The practical learning process of programming courses is divided into two stages: Prelab, which involves assignments before class, and Inlab, which involves assignments during direct classroom instruction. Both stages are conducted online, and student assignment data is recorded. This article suggests using features such as result, attempts, growths, Prelab, and Inlab. These features vary across model training iterations. They will be sequentially incorporated into the model; for example, if our output is Inlab1, then the input features will include all Prelab1 assignments. Similarly, if our output is Inlab2, then the input will include Prelab1, Inlab1, and Prelab2. This process repeats for Inlabs. This helps enhance the flexibility and effectiveness of the prediction model. The proposed prediction model used is Random Forest.

The integration of data from self-study and classroom assignments to predict future learning outcomes is an effective and promising method. Using a grading system based on academic performance as input for the model helps standardize and easily evaluate student performance.

The problem of Predicting Student Learning Performance using Machine Learning methods plays an important role in researching and improving the teaching and learning process

in modern higher education. We will apply the proposed model to real-world environments to serve students. We will integrate additional prediction models to provide appropriate warnings to individual learners. Additionally, we will strive to improve the model for higher accuracy. Furthermore, we hope to develop other related research areas such as:

• Using Machine Learning and Data Mining methods to analyze data from the LMS platform. These algorithms can be applied to predict student learning performance based on variables such as learning history, platform activities, and test results.

• Building prediction and evaluation models based on data to identify factors influencing student learning performance. These factors may include study time, interaction with learning content, and participation in learning activities.

• Developing intelligent interaction systems to provide personalized learning suggestions and instant feedback to students on the LMS platform. These systems can help students enhance their learning performance and commitment to the learning process.

• Conducting deeper analysis of student learning behaviors on the LMS platform to better understand how they interact with learning content and the challenges they face in the learning process.

In summary, researching and applying evaluation and prediction methods for student learning performance on the LMS platform not only benefits students but also helps improve the quality of education.

REFERENCES

[1] D. D. Angeline, P. Ramasubramanian, S. P. James I, & S. Hariharan. "Identification of learners' emotions in a learning environment using naïve bayes algorithm and evaluation of academic achivement with random forest AI", *International Journal of Information Retrieval Research (IJIRR),* vol. 12, no. 1, pp. 1-16, 2022.

[2] Y. E. Cakra and B. D. Trisedya, "tock price prediction using linear regression based on sentiment analysis," in *2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2015, pp. 147–154.

[3] S.A.KharyaShwetaandS.Soni,"Naivebayesclassifiers:Apr obabilistic detection model for breast cancer," *International Journal of Computer Applications*, vol. 92, no. 10, pp. 0975–8887, 2014.

[4] V. J. P. Superby J. F. and M. N., "Determination of factors influencing the achievement of the first-year university students using data mining methods," in *Workshop on Education*, 2006.

[5] A. A., "Monitoring student retention in the open university: Detritions, measurement, interpretation and action," *Open Learning*, vol. 19, no. 1, pp. 65–78, 2014.

[6] B. P. Gama J, "Characterization of classification algorithms," in *Progress in artificial intelligence: 7th Portuguese conference on artificial intelligence, EPIA'95 Funchal, Madeira Island, Portugal, October 3–6, 1995 Proceedings 7*, 1995, pp. 189–200.

[7] P. Balaji, S. Alelyani, A. Qahmash, and M. Mohana, "Contributions of machine learning models towards student academic performance prediction: a systematic review," *Applied Sciences*, vol. 11, no. 21, p. 10007, 2021.

[8] H. R. Hasan, A. S. A. Rabby, M. T. Islam, and S. A. Hossain, "Machine learning algorithm for student's performance prediction," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 2019, pp. 1–7.

[9] N. M. T. N. T. Uyen, "Predicting student's academic performance by applying data mining technique," *Journal of science - Vinh University*, vol. 48, no. 3A/2019, pp. 68–73, 2019.

[10] M. Pojon, "Using machine learning to predict student performance," Master's thesis, 2017.

[11] S. Hussain and M. Q. Khan, "Student-performulator: Predicting students' academic performance at secondary and intermediate level using machine learning," *Annals of data science*, vol. 10, no. 3, pp. 637–655, 2023.

[12] V. Matzavela and E. Alepis, "Decision tree learning through a predictive model for student academic performance in intelligent m-learning environments," *Computers and Education: Artificial Intelligence*, vol. 2, p. 100035, 2021.

[13] S. Kotsiantis, "Educational data mining: a case study for predicting dropout-prone students," *International Journal of Knowledge Engineering and Soft Data Paradigms*, vol. 1, no. 2, pp. 101–111, 2009.

[14] R. S. . O. S. Thien-Wan AU, "Performance prediction of learning programming - machine learning approach," in *Proceedings of the 30th International Conference on Computers in Education, Asia-Pacific Society for Computers in Education*, 2022, pp. 1–10.

[15] V. A. B. Jose Llanos and F. Restrepo-Calle, "Early prediction of student performance in cs1 programming courses," *PeerJ Computer Science*, vol. 1655, pp. 1–29, 2023.