Automation Process for Learning Outcome Predictions

Minh-Phuong Han[1], Trung-Tung Doan[2]*, Minh-Hoan Pham[3], Trung-Tuan Nguyen[4]
Thuongmai University, Hanoi, Vietnam[1]
FPT University, Greenwich Vietnam, Hanoi, Vietnam[2]
National Economics University, Hanoi, Vietnam[3,4]

***Abstract*—This paper presents a comprehensive study on the evaluation of algorithms for automating learning outcome predictions, with a focus on the application of machine learning techniques. We investigate various predictive models (logistic regression, random forest, gaussian naive bayes, k-nearest neighbors and support vector regression) to assess their efficacy in forecasting student performance in educational settings. Our experimental approach involves the application of these models to predict the outcomes of a specific course, analyzing their accuracy and reliability. We also highlight the significance of an automation process in facilitating the practical application of these predictive models. This study highlights the promise of machine learning in advancing educational assessment and paves the way for further investigations into enhancing the adaptability and inclusivity of algorithms in various educational settings.**

***Keywords*—Machine learning; predictive learning outcomes; education; logistic regression; k-nearest neighbors; Gaussian Naive Bayes; Random Forest; support vector regression**

I. Introduction

In the evolving landscape of educational technology, machine learning (ML) emerges as a pivotal tool, revolutionizing decision-making processes across various domains, particularly in education [1]. The integration of ML in educational settings has led to significant advances, such as personalized learning paths based on student data analysis, automated grading systems, and predictive models for student performance and learning outcomes [2]. These innovations underscore the transformative potential of ML in enhancing educational effectiveness and efficiency.

Despite these advances, the field faces several challenges. The complexity of educational data, characterized by its multidimensionality and the dynamic nature of learning processes, presents a significant hurdle [3]. Traditional ML algorithms often struggle to capture the nuanced patterns of learning, leading to inaccuracies in outcome predictions. Furthermore, the ethical considerations surrounding data privacy and the potential biases in algorithmic decisions add layers of complexity to the deployment of ML in education.

A critical overview of ML algorithms reveals their potential in predicting learning outcomes. These algorithms range from traditional statistical models to advanced deep learning networks, each offering unique perspectives in understanding student performance [2]. The importance of automation in this context cannot be overstated. Automation, in its essence, transforms the labor-intensive and often subjective process of outcome prediction into an objective, efficient, and scalable task [4].

An unsolved problem in this domain is the comprehensive automation of learning outcome predictions. While some progress has been made, existing systems either require significant manual intervention or fail to adapt to the evolving educational landscapes [1]. This gap not only hinders the scalability of ML solutions in education but also limits the potential for real-time, adaptive learning interventions.

The importance of this study lies in its focus on addressing these challenges by proposing an innovative approach to automate learning outcome predictions. By leveraging the latest advancements in deep learning and data analytics, this paper aims to develop a model that can accurately predict learning outcomes across diverse educational settings, thereby facilitating more personalized and effective learning experiences.

The novelty of our work is twofold. First, it introduces a novel algorithmic framework that combines the strengths of deep learning with the insights gained from educational psychology, aiming to better understand and predict learning behaviors. Second, it proposes a scalable automation process that can adapt to different learning environments and student profiles, significantly reducing the need for manual data processing and intervention.

This paper is structured into five sections, each designed to build upon the last in addressing the identified research gap. Following this introduction, we delve into a detailed review of the advances and challenges in ML applications in education, setting the stage for our novel contributions. We then present our methodology, focusing on the design and implementation of our predictive model. This is followed by an analysis of the results, demonstrating the effectiveness and adaptability of our approach. The paper concludes with a discussion of the implications of our findings for the future of ML in education, highlighting potential directions for further research.

II. Predicting Algorithms

A. Logistic Regression

Logistic Regression, a cornerstone in the realm of predictive analytics, offers a robust mathematical framework particularly suited for educational data [5]. Its essence lies in modeling the probability of a binary outcome, making it an ideal candidate for deciphering the dichotomous nature of learning outcomes: success or failure, pass or fail [6].

The application of Logistic Regression in predicting learning outcomes involves a meticulous process of mapping input variables — typically, the learning outcomes of certain subjects

— to a binary output, representing the predicted success or failure in other subjects. In the context of predicting learning outcomes, Logistic Regression is applied by modeling the probability of a student achieving a certain outcome (e.g., passing a subject) based on their performance in other subjects. The model's prowess stems from its ability to handle categorical data, a common characteristic of educational datasets. Logistic Regression shines in its simplicity and interpretability, a crucial aspect when educators and policymakers are at the helm, making decisions based on its predictions [7].

Several studies have illuminated the efficacy of Logistic Regression in educational settings. Singh and Jaiswal explored various machine learning classifiers, including Logistic Regression, in analyzing student performance in virtual learning environments [6]. Similarly, Lin et al. employed Logistic Regression, among other algorithms, to predict student submission timeliness in programming courses, highlighting the algorithm's versatility [8]. In conclusion, Logistic Regression stands out as a versatile and easily interpretable tool, essential for enhancing educational strategies through data analysis.

B. Random Forest

Random Forest, an ensemble learning method renowned for its robustness and accuracy, stands as a paragon in the domain of predictive analytics, particularly in educational settings [9]. At its core, Random Forest builds multiple decision trees and merges them to obtain a more accurate and stable prediction, a method especially effective in handling the multifaceted nature of educational data.

The Random Forest model can be conceptualized [10] as an aggregation of predictions from multiple decision trees, each contributing to the final decision. This ensemble approach significantly reduces the risk of over-fitting, a common pitfall in complex datasets such as educational data. The mechanics of Random Forest are particularly suited for educational data, which often encompasses a mix of categorical and continuous variables. By constructing a 'forest' of decision trees, each analyzing a subset of the data, Random Forest captures complex, non-linear relationships that might elude simpler models [11].

Several studies underscore the efficacy of Random Forest in this realm. Petkovic et al. demonstrated the algorithm's capability in predicting student learning effectiveness in software engineering teamwork with over 70% accuracy [9]. Su et al. applied Random Forest, among other algorithms, to predict student submission timeliness, showcasing its versatility in different educational scenarios [10]. Random Forest proves to be a vital and comprehensive tool for predicting educational outcomes, adept at managing complex datasets and offering interpretable insights for advanced learning strategies.

C. Gaussian Naive Bayes

Gaussian Naive Bayes, a probabilistic classifier underpinned by Bayes' Theorem, is a pivotal tool in the predictive analytics arsenal, particularly in the educational sector [12]. This algorithm stands out for its application of Gaussian probability distribution to handle continuous data, a common characteristic in educational datasets.

The application of Gaussian Naive Bayes in predicting learning outcomes involves a nuanced approach. It models the likelihood of outcomes based on input features, which in this context are the learning outcomes of selected subjects. The algorithm assumes that the features follow a Gaussian (normal) distribution, an assumption that simplifies the computation of probabilities. The strength of Gaussian Naive Bayes in educational settings lies in its ability to handle large datasets efficiently and its robustness in dealing with uncertainty in data. Its simplicity and the probabilistic basis provide a clear understanding of how predictions are made, which is crucial in educational contexts where interpretability is as important as accuracy.

Several studies have demonstrated the effectiveness of Gaussian Naive Bayes in educational data analysis. Wijaya et al. [12] applied the Naive Bayes algorithm to predict student success rates in learning, achieving high accuracy. Ouissal Sadouni and Abdelhafid Zitouni [13] discusses the implementation of dynamic optimization of learning indicators using Naive Bayes Classifier, which is relevant to understanding the application of Gaussian Naive Bayes in educational settings. Gaussian Naive Bayes stands out for its simplicity, efficiency, and effectiveness in analyzing complex educational datasets, making it a crucial tool for educational data analytics.

D. K-Nearest Neighbors

The K-Nearest Neighbors (KNN) algorithm, a cornerstone in the realm of machine learning, is renowned for its simplicity and effectiveness in classification and regression tasks. This non-parametric method operates on the principle that similar instances are likely to be found in close proximity [14].

In the context of predicting learning outcomes, KNN's application involves using the learning outcomes of certain subjects as input to predict the outcomes of others. The algorithm identifies the 'k' nearest data points to a query point and predicts the outcome based on the majority vote of these neighbors. The choice of 'k' and the distance metric, typically Euclidean, are crucial in this process. KNN's applicability in predicting learning outcomes is attributed to its ability to adapt to the intrinsic structure of educational data, which often exhibits complex, non-linear relationships. Its model-free nature allows for a flexible approach to understanding and predicting educational outcomes [15].

Several studies have underscored the utility of KNN in educational settings. Hendrianto et al. utilized KNN, among other algorithms, to predict student performance in compulsory subjects, demonstrating its predictive power in academic environments [14]. Tribhuvan and Bhaskar explored machine learning techniques, including KNN, to enhance student learning experiences, further highlighting the algorithm's relevance in educational data analysis [15]. KNN excels in educational data analysis with its simple implementation and local data-based predictions, showing promise as a tool for learning outcome predictions despite challenges like data scale sensitivity.

E. Support Vector Regression

Support Vector Regression (SVR), an extension of the Support Vector Machine (SVM) algorithm, is a powerful tool in the domain of machine learning, particularly for regression tasks. SVR is designed to find a function that approximates

the relationship between input and output variables in a high-dimensional space, making it suitable for complex prediction tasks [16].

In educational data analysis, SVR can be employed to predict learning outcomes. The algorithm takes as input the learning outcomes of selected subjects and predicts the outcomes of other subjects. The core of SVR lies in constructing a hyperplane in a multidimensional space that best fits the data points. The efficacy of SVR in predicting learning outcomes is attributed to its robustness against overfitting and its capacity to handle high-dimensional data.

Studies such as those by Pimentel et al. have demonstrated the application of SVR in educational settings, showcasing its potential in efficiently predicting student performance based on large datasets [16]. Another study by Huan Xu [17] introduces an innovative method for forecasting students' academic performance, which involves utilizing support vector regression (SVR) and enhancing it through the application of an improved dual algorithm.

III. Automation Process

The process of forecasting based on learner data is a multifaceted and intricate endeavor, requiring a harmonious integration of various stages including data collection, meticulous analysis, and the strategic application of advanced analytical techniques. In pursuit of optimizing this process, we propose a comprehensive and automated approach, encompassing a series of well-defined and interconnected steps. This automation process is not just a linear progression of tasks but a dynamic framework designed to adapt and evolve in response to the changing educational landscape and the diverse needs of learners. The automation process includes of following steps is illustrated in Fig. 1

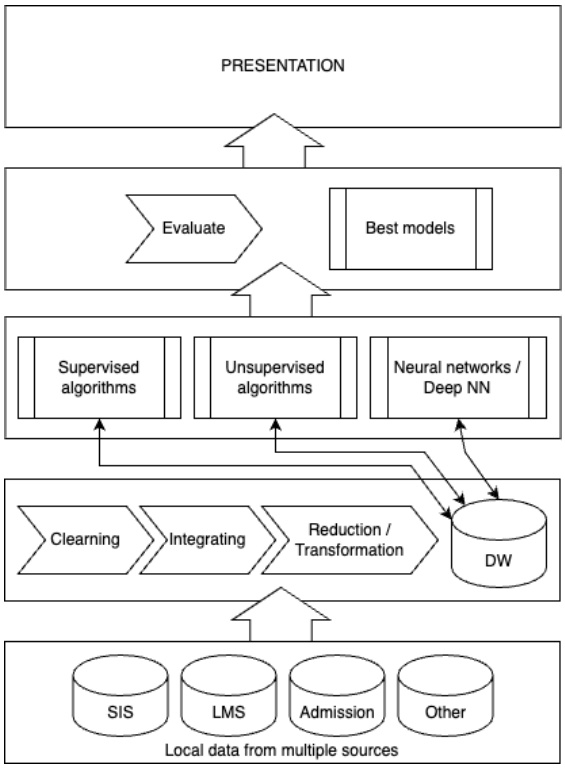

Fig. 1. Automation process.

1) Data collection: Learner data is amassed from various sources since their enrollment in university programs. This includes enrollment data like high school grades, English proficiency certificates, SAT scores, and aptitude assessments; and ongoing academic data such as grades, class attendance frequency, study hours, extracurricular activities, and more. This phase also involves identifying the most significant variables for forecasting purposes. The Student Information System (SIS) is a key data repository, storing demographic information (age, gender, nationality) and academic performance. However, socio-economic characteristics are not typically available in SIS, as they are often gathered through data collection methods like questionnaires. Additionally, learner information can be collected through Learning Management Systems (LMS) usage, including course data, grades, participation in discussions, and online exams and assignments.

2) Data preparation / preprocessing: Preparing data is crucial in data mining and involves making raw data suitable for mining techniques. Educational databases are often large, and the stored data frequently encounters quality issues. Hence, data cleansing methods are essential to handle missing, inconsistent, and outlier data to ensure data quality. Essential preprocessing methods include data cleaning, integration, reduction, and transformation.

- Data Cleaning involves removing noise and handling missing values to improve data quality.
- Data Integration combines data from multiple sources into a single source, addressing redundancy and inconsistency.
- Data Reduction transforms large datasets into smaller, information-rich datasets.
- Data Transformation modifies data into a form suitable for mining, including normalization and discretization.

Data after these steps are stored in a Data Warehouse (DW) which can be easily accessed for applying ML models in the next step.

3) Training ML models: This step involves using ML algorithms to analyze large data sets and identify patterns or trends. Both supervised and unsupervised learning models are employed to uncover interesting patterns in the data. Supervised learning uses labeled datasets to train models, which can then make predictions or classifications on new, unlabeled data. Unsupervised learning, on the other hand, involves analyzing data without any labels, aiming to identify patterns or clusters that can provide insights or aid decision-making. Classification and regression are two primary supervised learning techniques used for forecasting. Many more models/algorithms can be implemented and integrated into the system to provide flexibility in evaluating and selecting the best model for forecasting.

4) Model evaluation: Evaluating the performance of a classification model is a crucial step in developing and refining machine learning models. It allows for assessing the model's accuracy on test data. The original dataset is typically divided into two or three independent parts: a training set (validation/testing set) and a test set. The training set is used to

build the model, the test set to evaluate its performance, and in the case of large data, a validation set to optimize hyper parameters. Common methods for dividing the dataset include holdout, random sampling, and cross-validation. The results of running models / algorithms in the previous steps are evaluated to select the best model for each specific task.

5) Presentation: The final step in the automation process involves meaningfully and understandably presenting the results and findings from the selected models. The presentation step aims to convey the insights gained from the data mining process to stakeholders like educators, administrators, policymakers, and researchers in a format that supports decision-making and action.

IV. Experiment on Automation Process

We implement the automation process on a BI system described in our last paper [18]. The architecture of the BI system is shown in Fig. 2.

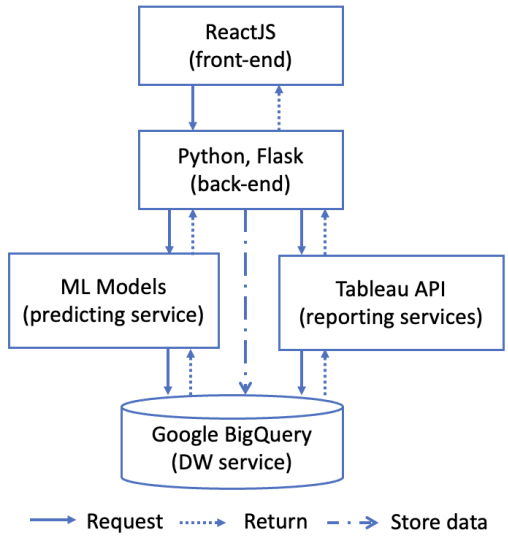

Fig. 2. The BI system.

In the back-end component, the steps of Data Collection and Data Preparation are executed seamlessly. This is followed by the Training ML models and the Model Evaluation steps, which are integral parts of the ML Models component. Subsequently, the Presentation step takes place in the front-end component. This component is a sophisticated web interface, designed to facilitate interaction with users through a web browser, ensuring a smooth and engaging user experience.

Using Data Collection and Data Preparation steps in the BI system, the authors have been updated the dataset to include data from years of 2022 and 2023, encompassing over 113,000 records. This dataset includes grades of students from three departments: Computing, Business Administration, and Graphic Design. In this experiment, the authors focused on the grades of Computing department.

A. Problem

Forecasting the academic performance in a subject based on the grades of previous subjects is a common and useful problem in the field of education. This not only helps students understand their abilities and developmental directions more clearly but also assists teachers and administrators in identifying and improving teaching methods as well as managing educational quality.

In the preceding discussion, this research will employ a dataset comprising grades from the Computing major to conduct experimental analyses. Specifically, a second-year course, designated as Advanced Programming (course code 1651), has been chosen as the focus for predicting academic outcomes namely, whether students pass or fail. This prediction will be based on the performance in a suite of first-year courses, which include Procedural Programming (1618), Programming (1619), Database Design & Development (1622), Website Design & Development (1633), Security (1623), and Managing a Successful Computing Project (1625). This approach allows for a detailed examination of the correlation between early academic performance and subsequent success in advanced coursework.

Following the data preparation phase, a selected subset of the requisite courses, encompassing the grades of 654 students, will be utilized for training and testing the predictive models. Within this subset, it is observed that approximately 80.6% of the students successfully passed the focal course 1651. Regarding the other courses integral to the prediction process, the average grades fluctuate, with the lowest mean grade being approximately 5.17 for course 1633 and the highest at around 5.93 for course 1619. Notably, course 1633 also exhibits the lowest pass rate at 68.81%, whereas course 1619 demonstrates the highest pass rate at 82.57%. These variations in pass rates and mean grades across different courses provide a comprehensive framework for analyzing and predicting academic performance in the Computing major.

B. Methods

To select the most effective model for prediction, the authors implemented the algorithms discussed in Section II and ran them on the same dataset to evaluate their outcomes. The model demonstrating the most optimal results was then chosen for integration into the automated forecasting process.

To assess the models' performance, the authors employed the k-folds verification technique with k = 10. Each model was trained and evaluated on each fold, calculating metrics such as Mean Squared Error (MSE), true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) and finally their average values on all the folds are returned. The correctness of the models was then determined based on the aggregation of true positives and true negatives across the data of each fold.

For all models, the authors trained and tested them on the same sub dataset of all grades from Computing Department. The sub dataset extracted from the main dataset had to ensure the inclusion of grades from all the relevant input and output courses, with at least one assessment per course. The total size of this qualified dataset comprised 654 students, making it suitable for running the predictive models.

All models uses the same inputs and output as described below:

- Input $X_1, X_2, X_3, X_4, X_5, X_6$: The average grades of assessments in first-year courses with codes 1618, 1619, 1622, 1633, 1623, and 1625.
- Output Y: Pass or fail outcome of a second-year course with the code 1651.

Next, we will see the result of training and testing on each model.

C. Experimental Result

1) Logistic regression result: To predict the binary outcome (pass/fail) of a second-year course (code 1651) using logistic regression we define the dependent variable, Y, represents the outcome of the course, coded as 1 for pass and 0 for fail. The independent variables, $X_1, X_2, X_3, X_4, X_5, X_6$, correspond to the average grades in six first-year courses with codes 1618, 1619, 1622, 1633, 1623, and 1625.

The logistic regression model is formulated as follows:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(w_0 + w_1X_1 + w_2X_2 + \ldots + w_6X_6)}} \quad (1)$$

where, $w_0, w_1, \ldots, w_6$ are the model parameters that need to be learned.

The model is trained by minimizing the logistic loss function defined as:

$$J(w) = -\frac{1}{6}\sum_{i=1}^{6}\Big[y_{(i)} \log\big(\sigma(w \cdot X_{(i)} + b)\big) + (1 - y_{(i)}) \log\big(1 - \sigma(w \cdot X_{(i)} + b)\big)\Big] \quad (2)$$

where, σ denotes the logistic (sigmoid) function.

For prediction, the model estimates the probability of a student passing the 1651 course. A student is predicted to pass (Y=1) if $P(Y = 1|X) > 0.5$; otherwise, the student is predicted to fail (Y=0).

The performance of the Logistic Regression model was evaluated using the k-fold cross-validation technique with $k = 10$ as described above. The results of k-folds verification, all values are average values of k-fold, are shown in Table I.

TABLE I. PERFORMANCE METRICS OF THE LOGISTIC REGRESSION MODEL

MSE	TN	FP	FN	TP	Correctness
0.17	4.2	3.0	2.0	20.8	83.33%

2) Random forest: To predict the binary outcome (pass/fail) of a second-year course (code 1651) using a Random Forest algorithm, the dependent variable, Y, is defined as the outcome of the course, coded as 1 for pass and 0 for fail. The independent variables, $X_1, X_2, X_3, X_4, X_5, X_6$, correspond to the average grades in six first-year courses with codes 1618, 1619, 1622, 1633, 1623, and 1625.

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification. The final prediction, $H(x)$, is the majority vote of the predictions made by individual trees, $h_1(x), h_2(x), \ldots, h_N(x)$, for N trees. In mathematical terms:

$$H(x) = mode\{h_1(x), h_2(x), \ldots, h_N(x)\} \quad (3)$$

Each tree is constructed using a random subset of the data, known as bootstrap sampling, and at each split in the tree, a random subset of the features is considered for splitting.

The Random Forest also uses out-of-bag (OOB) samples to estimate the error. The OOB error is the average error for each training sample, calculated using only the trees that did not have this sample in their bootstrap sample.

The effectiveness of the model was assessed through the application of the k-fold cross-validation method, wherein k was set to 10. The outcomes of this cross-validation, represented as mean values computed over all k-folds, are presented in Table II

TABLE II. PERFORMANCE METRICS OF THE RANDOM FOREST MODEL

MSE	TN	FP	FN	TP	Correctness
0.11	5.4	1.8	1.4	21.4	89.33%

3) Support Vector Regression (SVR): SVR is a type of Support Vector Machine (SVM) that is used for regression challenges. While traditional SVM is used for classification tasks, SVR can be used to predict continuous outcomes. The main idea behind SVR in our problem is to find a function $f(x) = w_1X_1 + w_2X_2 + w_3X_3 + w_4X_4 + w_5X_5 + w_6X_6 + b$ that has at most ϵ deviation from the actual target values Y for all the training data, and at the same time is as flat as possible.

Mathematically, SVR solves the following optimization problem:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \quad (4)$$

subject to

$$Y_i - (w_1X_{1i} + w_2X_{2i} + \ldots + w_6X_{6i} + b) \leq \epsilon + \xi_i, \quad (5)$$

$$(w_1X_{1i} + w_2X_{2i} + \ldots + w_6X_{6i} + b) - Y_i \leq \epsilon + \xi_i^*, \quad (6)$$

$$\xi_i, \xi_i^* \geq 0 \quad (7)$$

where, w is the weight vector, b is the bias, C is the regularization parameter, ξ and ξ^* are slack variables that allow for violations of the ϵ margin.

The efficacy of the SVR model was gauged utilizing the k-fold cross-validation approach, setting k at 10. The outcomes from the k-folds assessment, which are the mean values calculated across all k-folds, are outlined in the Table III.

TABLE III. PERFORMANCE METRICS OF THE SVR MODEL

MSE	TN	FP	FN	TP	Correctness
0.133	4.6	2.6	1.4	21.4	86.67%

4) K-Nearest neighbors: The KNN algorithm operates by identifying the 'K' nearest neighbors of a given data point in the feature space. The Euclidean distance is commonly used as the distance metric, calculated as $d(X_i, X_j) = \sqrt{\sum_{n=1}^{N}(X_{in} - X_{jn})^2}$, where X_i and X_j are two points in an N-dimensional space.

In this application of the KNN algorithm, each student's likelihood of passing or failing the second-year course (code 1651) is predicted based on the outcomes of the nearest neighbors in the dataset. These neighbors are identified by comparing the average grades in six other courses (codes 1618, 1619, 1622, 1633, 1623, and 1625) of each student. For a given student, the algorithm locates the 'K' students most similar in terms of their first-year grades and predicts the student as likely to pass (Y=1) or fail (Y=0) the 1651 course based on the most common outcome among these 'K' nearest neighbors. This can be represented as $Y = mode\{c_1, c_2, ..., c_K\}$, where c_i is the pass/fail outcome of each neighbor. Furthermore, a weighted voting approach can be employed where the influence of each of the 'K' neighbors on the prediction is inversely proportional to their grade distance from the student being classified, giving closer students (more similar in terms of grades) a higher influence in the prediction.

The KNN model's effectiveness in forecasting the outcome of course 1651 was appraised through the k-fold cross-validation method, employing $k = 10$. Table IV shows the ensuing results represent the average values derived from the k-folds:

TABLE IV. PERFORMANCE METRICS OF THE KNN MODEL

MSE	TN	FP	FN	TP	Correctness
0.18	3.0	4.2	1.2	21.6	82.00%

5) Gaussian Naive Bayes (GNB): In the context of the Gaussian Naive Bayes (GNB) model, we aim to predict the probability $P(Y = 1|X)$, which represents the probability of a student passing the course 1651 (coded as 1 for pass) given a set of relevant grades of other courses represented by the feature vector X.

- Y: A binary variable representing the result of 1651 course (pass or fail)
- X: A feature vector representing the grades $X_1, X_2, \ldots, X_6$, where X_i represents the grade in the respective course i.

The GNB model calculates the probability $P(Y = 1|X)$ using Bayes' theorem, which relates the conditional probability $P(Y = 1|X)$ to the joint probability $P(X, Y)$ and the marginal probability $P(X)$:

$$P(Y = 1|X) = \frac{P(X|Y = 1) \cdot P(Y = 1)}{P(X)} \tag{8}$$

In this equation:

- $P(Y = 1|X)$: The probability of passing the course given the feature vector X
- $P(X|Y = 1)$ The probability distribution of the feature vector X when the student passes the course
- $P(Y = 1)$ The prior probability of passing the course
- $P(X)$ The marginal probability of observing the feature vector X

The GNB model assumes that each feature X_i follows a Gaussian distribution for each class (pass or fail). It calculates these probabilities based on training data and assumes that features are conditionally independent given the class label. In summary, the GNB model uses Bayes' theorem and Gaussian distributions to estimate the probability of a student passing the course based on grades in other courses. It's trained on labeled data to estimate Gaussian distribution parameters, including mean and variance, for each feature in both pass and fail classes.

The effectiveness of the GNB model in predicting the results of course 1651 was assessed using the k-fold cross-validation approach, with k set to 10. Presented in the Table V are the aggregated average results from these k-folds.

TABLE V. PERFORMANCE METRICS OF THE GNB MODEL

MSE	TN	FP	FN	TP	Correctness
0.18	6.0	1.2	4.1	18.7	82.33%

D. Experiment Result Analysis

In this section, we present and analyze the results obtained from the k-fold cross-validation (with $k = 10$) for various predictive models: Logistic Regression, Random Forest, SVR, KNN, and GNB.

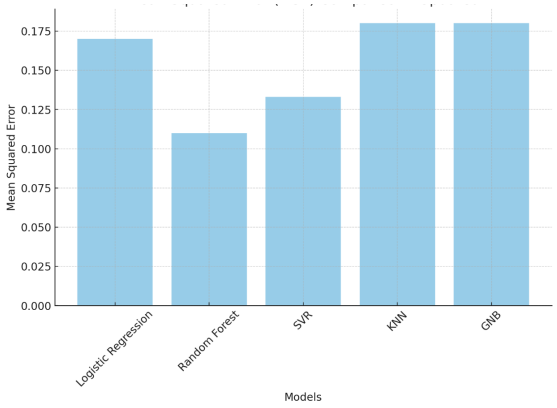

Fig. 3. Mean Squared Error (MSE) comparison.

1) Mean squared error comparison: As shown in Fig. 3, the Random Forest model demonstrated the lowest MSE (0.11), suggesting it as the most accurate among the evaluated models. Conversely, both KNN and GNB models exhibited the highest MSE (0.18), indicating relatively higher prediction errors.

2) Classification results: Classification results, including True Negatives (TN), False Positives (FP), False Negatives (FN), and True Positives (TP) counts, are essential for understanding a model's capability in correctly classifying different

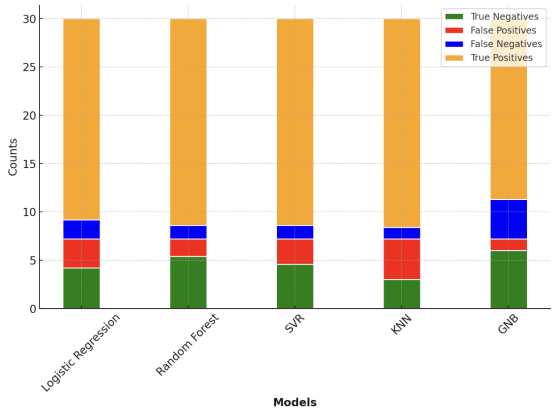

Fig. 4. Classification comparison.

outcomes. Fig. 4 illustrates a comparative analysis of these metrics. Notably, the Gaussian Naive Bayes (GNB) model excelled in identifying negative cases (TN), while the K-Nearest Neighbors (KNN) model slightly led in identifying positive cases (TP). However, the Random Forest model balanced false positives and false negatives effectively, indicating robust classification capabilities.

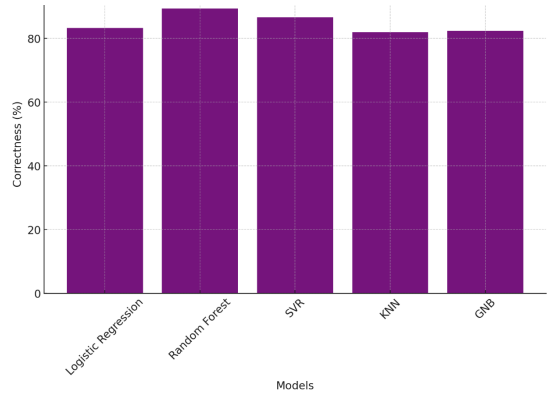

Fig. 5. Correctness comparison.

3) Correctness of models: The correctness percentage provides an overall effectiveness measure of the models. As depicted in Fig. 5, the Random Forest model outperformed others with the highest correctness percentage (89.33%). Despite high TP and TN rates in KNN and GNB models, respectively, their overall correctness percentages were lower, suggesting a trade-off between different types of classification errors.

4) Integration of predictive model: The analysis reveals that the Random Forest model exhibits a balanced and superior performance across various metrics, making it the most effective model among those tested. The GNB model, despite its high rate of TN, struggles in overall performance, indicating a potential issue in classifying positive cases. The KNN model, while showing a high TP rate, suffers from high MSE and lower correctness, pointing towards possible overfitting or poor generalization.

These insights suggest that further refinement and tuning of the Random Forest model could yield even better results.

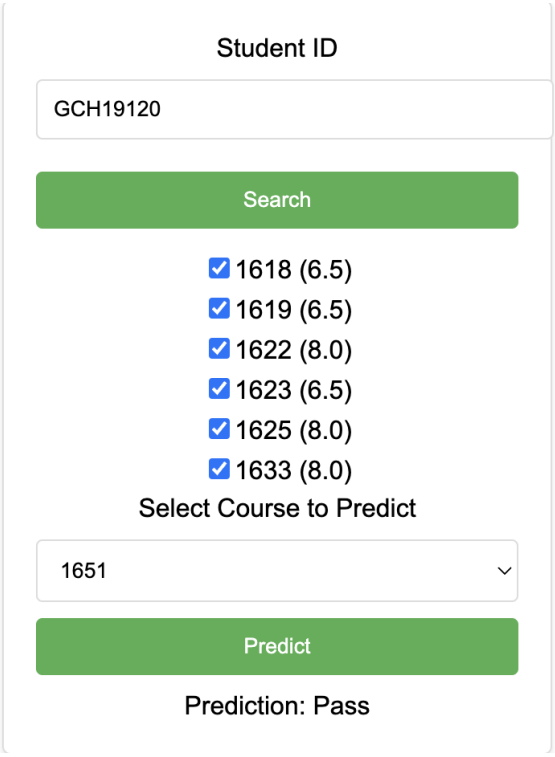

Fig. 6. Predict 1651 course result.

Additionally, a deeper investigation into the feature selection and parameter optimization for the GNB and KNN models might improve their performance.

Based on the analysis, we integrate the Random Forest model into the Automation process. With a simple Web interface in the Presentation step, it allows user to predict the result of a course according to the grades of other selected courses as in Fig. 6.

We also did another experiment in which we select fewer courses. Since 1651 is a programming course (name: Advanced Programming), we select only programming related courses which are 1618 (Programming), 1622 (Database Design & Development) and 1633 (Website Design & Development). The result is shown in the Table VI.

TABLE VI. PERFORMANCE METRICS OF VARIOUS MODELS

Model	MSE	TN	FP	FN	TP	Correctness
RandomForest	0.133	5.1	2.1	1.9	20.9	86.67%
SVR	0.137	4.7	2.5	1.6	21.2	86.33%
KNN	0.163	4.1	3.1	1.8	21.0	83.67%
GaussianNB	0.163	6.0	1.2	3.7	19.1	83.67%
LogisticRegression	0.150	5.2	2.0	2.5	20.3	85.00%

The results indicate that the Random Forest model consistently achieves the lowest MSE and the highest level of accuracy. However, it's noteworthy that the accuracy of the Random Forest model shows a slight decline compared to its performance when trained with a more comprehensive dataset that includes both programming and theoretical courses. This observation raises an intriguing question: Does the inclusion of a broader selection of previous courses enhance the predictive

accuracy of a model? Interestingly, this does not seem to be the case for other models like Logistic Regression or Gaussian Naive Bayes, suggesting that the relationship between the breadth of course selection and predictive accuracy is not straightforward and may vary across different ML models.

This further confirms the necessity of an automated process that involves implementing various ML algorithms to allow users to choose the best model after the evaluation step. Users can select different models/algorithms for different forecasting problems or even different models/algorithms for different subjects in a specific learning outcome forecasting problem.

V. Conclusion

Our study embarked on an journey to unravel the potential of integrating various machine learning models in an automation process to predict educational outcomes. The heart of our exploration was the rigorous experiment that tested models like Logistic Regression, Random Forest, KNN, and Gaussian Naive Bayes against the challenging task of forecasting course results.

The findings are illuminating. The Random Forest model, in particular, demonstrated exceptional proficiency, marked by the lowest MSE and highest correctness in predictions. This underscores its potential as a reliable tool in educational settings. Moreover, our analysis revealed an intriguing trend: the accuracy of predictions increases with the inclusion of more previous course grades. This insight is pivotal for educational institutions aiming to leverage data-driven approaches for student assessment and support.

Our study also emphasized the importance of a user-friendly interface in the Presentation stage, allowing educators and stakeholders to seamlessly interact with the predictive models. The practical application of our research, illustrated through a simple web interface, bridges the gap between complex algorithms and real-world usability.

In conclusion, this research marks a significant stride towards integrating machine learning in educational technology. It not only sheds light on the efficacy of various predictive models but also paves the way for future investigations. Areas ripe for exploration include enhancing model robustness and exploring their adaptability across diverse educational contexts. As we tread into this future, our endeavor remains rooted in the goal of harnessing technology to enrich learning experiences and outcomes.

Acknowledgment

We are grateful for the opportunity to conduct this research within the framework of the Project BIG - BI for Greenwich at FPT University (Greenwich Vietnam). Our thanks extend to the university for providing access to real anonymized data, which was crucial for this study.

References

[1] P. Balaji, Salem Alelyani, Ayman Qahmash, and Mohamed Mohana. Contributions of machine learning models towards student academic performance prediction: A systematic review. *Applied Sciences*, 11(21), 2021.

[2] Areej M. Alhothali, Maram Albsisi, H. Assalahi, and T. Aldosemani. Predicting student outcomes in online courses using machine learning techniques: A review. *Sustainability*, 14(10), 2022.

[3] Worawat Lawanont and Anantaya Timtong. Smart education using machine learning for outcome prediction in engineering course. In *2022 14th International Conference on Knowledge and Smart Technology (KST)*, 2022.

[4] Narcisa Roxana Mosteanu. Machine learning and robotic process automation take higher education one step further. Online, Accessed: 2023.

[5] V. Uskov, J. Bakken, Adam Byerly, and Ashok Shah. Machine learning-based predictive analytics of student academic performance in stem education. In *IEEE Global Engineering Education Conference (EDUCON)*, 2019.

[6] Neha Singh and U. C. Jaiswal. Analysis of student study of virtual learning using machine learning techniques. *International Journal of Synthetic Emotions (IJSE)*, 2022.

[7] Scott H. Yamamoto and Charlotte Y. Alverson. Outcomes of students with disabilities after exiting from high school: A study of education data use and predictive analytics. *Journal of School Leadership*, 2022.

[8] Yu-Sheng Su, Yu-Da Lin, and Tai-Quan Liu. Applying machine learning technologies to explore students' learning features and performance prediction. *Frontiers in Neuroscience*, 2022.

[9] D. Petkovic, S. Barlaskar, Jizhou Yang, and R. Todtenhoefer. From explaining how random forest classifier predicts learning of software engineering teamwork to guidance for educators. In *IEEE Frontiers in Education Conference (FIE)*, 2018.

[10] Yu-Sheng Su, Yu-Da Lin, and Tai-Quan Liu. Applying machine learning technologies to explore students' learning features and performance prediction. *Frontiers in Neuroscience*, 2022.

[11] Justine B Nasejje, R. Mbuvha, and H. Mwambi. Use of a deep learning and random forest approach to track changes in the predictive nature of socioeconomic drivers of under-5 mortality rates in sub-saharan africa. *BMJ Open*, 2022.

[12] B. A. Wijaya, Vijay Kumar, Berlian Fransisco Jhon Wau, J. Tanjung, and N. Dharshinni. Application of data mining using naive bayes for student success rates in learning. *Management and Business Innovation*, 2022.

[13] O. Sadouni and A. Zitouni. Task-based learning analytics indicators selection using naive bayes classifier and regression decision trees. In *2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, 2021.

[14] Raphael Kusumo Hendrianto, A. Siagian, and R. Alfanz. Using data mining to predict students' performance: A case study in sultan ageng tirtayasa university. *Setrum: Sistem Kendali-Tenaga-Elektronika-Telekomunikasi-Komputer*, 2022.

[15] R. R. Tribhuvan and T. Bhaskar. Machine learning techniques for enhancing student learning experiences. *Journal of Information Technology and Software Engineering*, 2021.

[16] J. S. Pimentel, R. Ospina, and Anderson Ara. Learning time acceleration in support vector regression: A case study in educational data mining. *Stats*, 4(3), 2021.

[17] Huan Xu. Prediction of students' performance based on the hybrid ida-svr model. *Complexity*, 2022, 2022.

[18] H.M. Phuong, P.M. Hoan, N.T. Tuan, and D.T. Tung. Predicting student study performance in a business intelligence system. In *Intelligent Systems and Networks. ICISN 2023. Lecture Notes in Networks and Systems*, volume 752. Springer, Singapore, 2023.