Techno-Science Research Journal V14 (Special Issue) (2026) P 28-39

Content list available at ITC

Techno-Science Research Journal

Journal Homepage: http://techno-srj.itc.edu.kh/

From Educational Model to Achievements - Classifying Post-Graduate Success with Machine Learning

Samboeun Hean[1,2*], Sokchea Kor[1], Tepmony Sim[2], Kimtho Po[2], Srun Sovila[3]

[1]*Cambodia Academy of Digital Technology, Phnom Penh, Cambodia*
[2] *Institute of Technology of Cambodia, Phnom Penh, Cambodia*
[3]*Royal University of Phnom Penh, Phnom Penh, Cambodia*

Received: 06 June 2025; Revised: 13 August 2025; Accepted: 12 September 2025; Available online: 31 July 2026

Abstract: *Student post-graduate success plays an essential role for higher education institutions (HEIs) to improve their standing and reputation, increasing their enrollment rates. As a result, HEIs are allocating significant effort and resources to cultivate student success after graduation. In this study, student post-graduate success is categorized into four distinct levels including "Failure", "Satisfaction", "Success", and "High Success". To predict these outcomes, we employed four widely recognized machine learning methods: Random Forest Classifier (RFC), Decision Tree Classifier (DTC), Logistic Regression Classifier (LRC), and Support Vector Machine Classifier (SVMC). Our dataset comprised 373 student records compiled from NIPTICT/CADT, encompassing six generations across the majors of Computer Science, Telecommunications and Networking and Digital Business. The data were organized into three distinct categories: student profile, educational model, and student success. To prepare this raw data for training and testing our machine learning algorithms, manual annotation was performed. The weight score for each student was calculated based on our proposed post-graduate success metric. This calculated weight score is then used to classify students into their respective success levels. The result showed that the DTC achieved the highest performance, outperforming other classifiers with an accuracy of 0.96 on the validation set and 0.98 on the test set. RFC followed, demonstrating an accuracy of 0.92 on the validation set and 0.93 on the test set. Conversely, LRC and SVMC showed comparatively lower performance, achieving accuracies of 0.88 and 0.8 on the validation set and 0.89 and 0.84 on the test set respectively. The unexpectedly high performance of the Decision Tree prompted deeper investigation. We performed k-fold cross-validation to both validate these results and explore the factors contributing to its surprising effectiveness. Through cross-validation, DTC exhibits a slightly higher mean and higher standard deviation than RFC. This substantially higher standard deviation suggests that DTC's performance is less consistent and potentially less reliable when classifying student post-graduate success in real-world application. When these models are deployed and retrained with new data, the Decision Tree could exhibit greater fluctuations in predictive accuracy or other performance metrics. In contrast, Random Forest typically mitigates individual tree biases and variances, leading to more stable overall performance. Consequently, the Random Forest model is more suitable for this dataset and student post-graduation success prediction.*

Keywords: Student Post-Graduate Success Classification, Post-Graduate Success Metric, Random Forest Classifier, Decision Tree, SVM, Logistic Regression

1. INTRODUCTION

Student post-graduate success is a key performance indicator for HEIs, directly influencing their standing and reputation within an increasingly competitive educational environment. Higher education institutions face ongoing challenges related to student achievement, driven by the desire to improve graduation rates, manage budget constraints tied to academic performance, meet accreditation standards, satisfy employer demands, and the need to remain competitive (Ayán & García, 2008). Therefore, HEIs are increasingly putting more effort and resources to ensure student success post-graduation. In this research, we defined student post-graduate success as the completion of their studies and securing relevant employment in their field of study with above-average salary. Determining post-graduate success of a student can be challenging due to its multifaceted nature, with perspectives varying among students, employers, and training institutions. In consultation with

* Corresponding author: Hean Samboeun
E-mail: samboeun.hean@cadt.edu.kh

companies from various sectors, encompassing from small, medium and large enterprises, post-graduate success is defined based on three criteria: **relevant job, starting salary, and employer satisfaction**. In this work, we will predict student post-graduate success using machine learning models: Random Forest, Decision Tree, Support Vector Machine, and Logistic Regression.

2. RELATED WORKS

Essa and Ayad (2012) tried to develop and implement the Student Success System (S3), utilizing predictive modeling and data visualizations to identify at-risk students and improve retention, completion, and graduation rates. They emphasize S3's comprehensive approach and its aim to overcome limitations of existing predictive models in education. Jayaprakash et al. (2015) predicted student performance at Blue-Crest College by applying Naïve Bayes, achieving 92.2% accuracy within two seconds and outperforming feedback-based predictions. Their findings highlight the potential of data mining for spotting struggling students early and assisting in institutional decision-making. Belachew and Gobena (2017) predicted student performance at Wolkite University using neural networks, Naïve Bayesian, and SVM. Their study found Naïve Bayes outperforming the other models (95.7%) for IT students, though they noted these methods' limited application in enhancing student learning. Oyerinde and Chia (2017) analyzed student academic performance using multiple linear regression, finding mathematics courses strongly influenced CS201 performance. They highlighted a conceptual gap due to conflicting results on optimal prediction models and a methodological gap regarding linear regression's reliability with small sample sizes. Oloruntoba and Akinode (2017) utilized Support Vector Machine (SVM) to predict student academic performance (GPA) at a Nigerian polytechnic. Their SVM model, trained on 'O' level and early semester results, achieved 97% prediction accuracy with an RBF kernel (C=100), outperforming KNN, decision trees, and linear regression. Gerritsen (2017) compared Neural Networks (NN) against six other classifiers for student performance prediction by utilizing Moodle log data. NN achieved the highest accuracy (66.1%), outperforming others like Logistic Regression (62.4%). Pojon (2017) explored student performance prediction using linear regression, decision trees, and Naïve Bayes, finding feature engineering more impactful than method selection for prediction accuracy improvement. While Naïve Bayes achieved 98% and Decision Trees 78% accuracy on different datasets. Ulinnuha et al. (2017) compared four models—Random Forest, Artificial Neural Network, Naïve Bayes, and Logistic Regression—for academic performance prediction. Neural Networks consistently performed better, while Random Forest suffered from overfitting. Naïve Bayes and Logistic Regression demonstrated similar performance, emphasizing the importance of training dataset composition. Obsie and Adem (2018) predicted student performance at Hawassa University utilizing Neural Network (NN), Linear Regression (LR), and Support Vector Regression (SVR), with SVR and LR found to be superior to NN. Despite data mining proving crucial for predicting graduation CGPA. Kim et al. (2018) introduced GritNet, an Artificial Neural Network architecture with an unsupervised domain adaptation method for student performance prediction. Their research demonstrated GritNet's ability to generalize across Udacity Nanodegree programs and improve early, real-time predictions without custom feature engineering or new labeled data, showcasing its high adaptability. Buenaño-Fernández et al. (2019) revealed machine learning's effectiveness in predicting computer engineering students' academic performance, identifying Decision Tree as most effective. Rajalaxmi et al. (2019) applied regression models to predict engineering student academic performance, demonstrating accurate predictions useful for educators in assessing class performance and refining teaching methods. They noted that multiple regression may lack accuracy with small sample sizes, presenting a methodological limitation. Vinod and Bhatt (2019) predicted postgraduate student performance using Artificial Neural Networks (ANN), achieving 97.429% accuracy and outperforming Linear Regression and Random Forest. They highlighted ANN's efficiency and generalization benefits. Sekeroglu, Dimililer, and Tuncal (2019) utilized machine learning to predict and classify student performance based on social and family factors across two datasets. Support Vector Regression (SVR) achieved highest prediction scores, while Backpropagation (BP) was best for classification (87.78% accuracy). Zohair & Mahmoud (2019) predicted student performance using small datasets with clustering and visualization. Support Vector Machine and Learning Discriminant Analysis proved efficient. Lau, Sun, and Yang (2019) predicted student academic performance using artificial neural networks (ANNs), achieving 84.8% accuracy. They noted, however, that it remains unclear whether ANNs can enhance learning outcomes. Anderson, Boodhwani, and Baker (2019) predicted student graduation at an R1 university using various ML models. This study found that stochastic gradient descent (SGD) classifiers are the most accurate (0.824). Baashar et al. (2022) conducted a study predicting student performance based on internal assessments, utilizing Artificial Neural Networks to achieve 95.34% accuracy.

3. DATASET DETAILS

The data utilized in this research was compiled from NIPTICT/CADT student records encompassing six generations across the majors of Computer Science, Telecommunications and Networking and Digital Business. Data was divided into three distinct categories: student profile, educational model, and student success, as depicted in Error! Reference source not found.. Each category comprised 373 samples. Student profile and educational model data were extracted directly from CADT's university management system, whereas student success data

was gathered from both students and employers using quantitative data collection techniques.

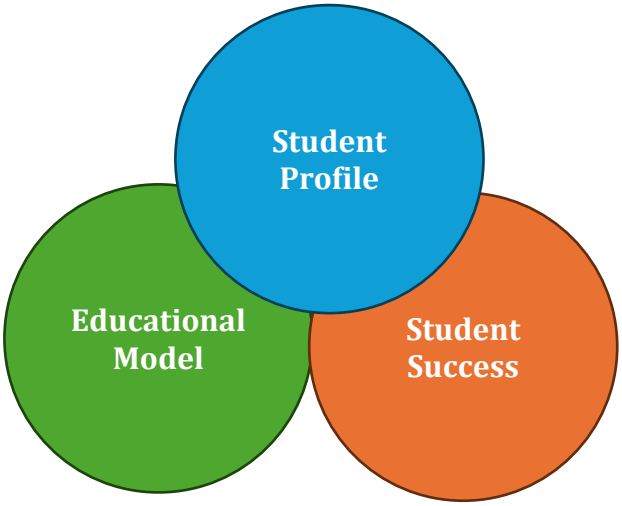

Fig. 1. Types of Data

3.1 Student Profile Data

In the student profile dataset, unimportant information was excluded from the research. We retained only the crucial features including ID, generation, gender, province/city, class, exam year, department, and major, as shown in

Table *1* shows a representative sample of six student profiles records from distinct generations, with one student profile randomly selected from each generation.

3.2 Educational Model Data

The educational model of each generation was recorded in order for evaluation. Similar to student profile data, only the essential features were retained. These features are ID, Elective Course (EC), Core Course (CC), Foundation Course (FC), Soft Skills, Internship, Capstone Project, Weekly Seminar (WS), Workshop, Industry Visit, Entrance Exam, Lecturer with Bachelor (LWB), Lecturer with Master (LWM), Lecturer with PhD (LWP), Teaching Assistant (TA), Internal Lecturer (IL), Industry Expert (IE), Inviting Professor (IP), Faculty-Student-Ratio (FSR), Volunteer Program (VP), as depict in **Table 2.**

3.3 Student Success Data

This dataset records the post-graduate success of students. It consists of key information such as Generation, Student ID,

Table *1*. The student ID is used to identify the student and will not be used to train the model.

Employer Satisfaction

This data is obtained from employer evaluations of student performance following the completion of internship programs at companies. Evaluations are collected from a diverse range of companies ranging from large, and medium, to small enterprises. The evaluation form requires employers to rate students on a scale of 1 to 5 across four key areas: technical skills, soft skills, problem-solving abilities, and personal attitude. Then, the individual ratings are aggregated, and the resulting overall score is normalized to a scale of 1 to 15. We categorize the employer satisfaction level for students as follows:

- Dissatisfied: Score below 7.5
- Neutral: Score between 7.5 and 10
- Satisfied: Score greater than 10 to 12.5
- Highly Satisfied: Score above 12.5

Starting Salary, Relevant Job, and Employer Evaluation. Data collection involved two surveys: one administered to students and another to their respective employers.

Starting Salary

Starting salary refers to the first remuneration received upon graduation. To gather this information, we employed a quantitative data collection method utilizing surveys. The starting salary questionnaire presented respondents with four options: 1) Below USD 300, 2) In range of USD 300 - 449, 3) In range of USD 450 - 599, and 4) From USD 600, as detailed in Table 4.

Relevant Job

CADT evaluates whether each generation of students secures employment aligned with their respective majors. This data serves as a crucial input for training institutions to refine the curricula. To gather this information, we employed a questionnaire within the survey, offering three response options: Yes, In Between, and No, as seen in **Table 4**. To validate the accuracy of the response, we also included a question about job position within the company. By analyzing responses to both sets

of questions, we concluded whether the student's employment aligns with their respective majors.

Employer Satisfaction

This data is obtained from employer evaluations of student performance following the completion of internship programs at companies. Evaluations are collected from a diverse range of companies ranging from large, and medium, to small enterprises. The evaluation form requires employers to rate students on a scale of 1 to 5 across four key areas: technical skills, soft skills, problem-solving abilities, and personal attitude. Then, the individual ratings are aggregated, and the resulting overall score is normalized to a scale of 1 to 15. We categorize the employer satisfaction level for students as follows:

- Dissatisfied: Score below 7.5
- Neutral: Score between 7.5 and 10
- Satisfied: Score greater than 10 to 12.5
- Highly Satisfied: Score above 12.5

Table 1. Student Profile Data

Generation	ID	Sex	Provice/City	Class type	Exam Year	Grade	Department
1	B20140004	M	Kompng Cham	Science	2014	C	Computer Science
2	B02EL009	M	Siem Reap	Social Science	2015	E	Digital Business
3	B20160037	F	Kompong Speu	Science	2016	B	Telecom & Networking
4	B20170039	M	Takeo	Science	2017	D	Computer Science
5	B20180029	F	Phnom Penh	Science	2018	A	Computer Science
6	B20190085	F	Stung Treng	Science	2019	D	Digital Business

Table 2. Educational Model Data

ID	EC (%)	CC(%)	FC(%)	Soft Skills(%)	Internship	CP	WS	Workshop	Industry Visit	Entrance Exam	LWB(%)	LWM(%)	LWP(%)	TA (%)	IL (%)	IE (%)	IP (%)	FSR	VP
B20140004	15	50	25	10	2	2	96	8	12	1	15	72	13	0	85	10	5	7	1
B02CS021	15	45	25	15	2	2	98	8	12	1	12	73	15	0	85	10	5	12	1
B20160013	15	45	22	18	2	2	98	8	12	1	10	73	17	0	80	15	5	13	0
B20170023	15	42	25	18	2	2	98	8	12	1	8	74	18	10	70	15	5	14	1
B20180038	20	40	25	15	2	2	100	10	12	1	5	75	20	15	65	14	6	15	0
B20190096	20	40	25	15	2	2	100	10	12	1	0	80	20	15	60	18	7	16	0

Table 3. Student Success Data

Generation	ID	Starting Salary	Relevant Job	Employer Satisfaction
1	B20140016	In range of USD 300 - 449	Yes	Highly Satisfied
2	B02EL047	Below USD 300	No	Satisfied
3	B20160014	From 600	Yes	Satisfied

4	B20170029	In range of USD 300 - 449	Yes	Highly Satisfied
4	B20170027	In range of USD 450 - 599	Yes	Highly Satisfied
5	B20180051	In range of USD 300 - 449	Yes	Highly Satisfied
6	B20190091	Below USD 300	In Between	Satisfied

3.4 Post-Graduate Success Metric

Defining post-graduate success can be challenging due to its multifaceted nature, with perspectives varying among students, employers, and training institutions. In this study, to define the criteria for post-graduate success, we conducted a closed-group discussion and a survey with companies from various sectors such as finance, telecommunication, and tech companies. We presented a list of potential graduate success criteria and asked companies to rate each criterion. A list of these criteria is presented below.

- Employment Rate
- Relevant Job
- Starting Salary
- Career Advancement
- Employer Satisfaction
- Life-Long Learning
- Personal Fulfillment
- Pursuit of Higher Degrees
- Job Placement Rate

Based on the survey results, companies consistently selected Relevant Job (RJ), Starting Salary (SS), and Employer Satisfaction (ES) as crucial factors for graduate success. Employer Satisfaction was ranked highest, followed by Starting Salary, and Relevant Job. These three factors will serve as the foundation for developing a comprehensive Graduate Success Metric. Each criterion will be assigned a respective score and coefficient, weighted according to its level of importance. You can see the score and coefficient of Relevant Job, Starting Salary, and Employer Satisfaction in **Table 4**, **Table 5**, **Table 6** respectively. According to the ranking, weights of 3, 2, and 1 were assigned to ES, SS, and RJ respectively.

Table 4. Relevant Job with its respective score and coefficient

Relevant Job	Score (S_1)	Coefficient (C_1)
No	1	1
In Between	2	
Yes	3	

Table 5. Starting Salary with its respective score and coefficient

Starting Salary	Score (S_2)	Coefficient (C_2)
SS < 300USD	1	2
300USD ≤ SS < 449USD	2	
500USD ≤ SS < 600USD	3	
SS > 600USD	4	

Table 6. Employer Satisfaction with its respective score and coefficient

Satisfaction Level	Score (S_1)	Coefficient (C_1)
Dissatisfied	1	3
Neutral	2	
Satisfied	3	
Highly Satisfied	4	

Weighted Score Equation

$$\boldsymbol{Weighted\ Score\ (WS)} = \sum_{i=1}^{3} \boldsymbol{C_i S_i} \qquad (1)$$

$\boldsymbol{i}$: is the index running from 1 to 3

$\boldsymbol{C_i}$: is the i^{th} coefficient

$\boldsymbol{S_i}$: is the i^{th} score

To determine four different classes, threshold values were defined as below:

T_1: is the threshold value of level 1

T_2: is the threshold value of level 2

T_3: is the threshold value of level 3

The values of T1, T2, and T3 are calculated from the Maximum Weighted Score (MWS). To derive MWS, we sum the products of each factor's maximum score and its corresponding coefficient.

We calculate the Maximum Weighted Score using the equation 1 as follows:

$$\text{MWS} = \sum_{i=1}^{3} \boldsymbol{C_i S_i} = C_1 S_1 + C_2 S_2 + C_3 S_3$$

$$= 2 * 4 + 1 * 3 + 3 * 4 = 23$$

$$T_1 = \frac{MWS}{2} = \frac{23}{2} = 11.5$$

To determine the values of Threshold T2 and T3, we need to calculate the interval value of each level.

$$\text{J} = \lceil (\text{MWS} - \text{T1})/3 \rceil = \lceil (23 - 11.5)/3 \rceil = 4$$

$$T_2 = T_1 + J = 11.5 + 4 = 15.5$$

$$T_3 = T_2 + J = 11.5 + 4 = 19.5$$

Table 7. Weighted Score Levels and Corresponding Classes

N°	WS	Class
1	WS < T1	Failure
2	T1 ≤ WS < T2	Satisfaction
3	T2 ≤ WS < T3	Success
4	WS >= T3	High Success

3.5 Data Annotation

To prepare data for training and testing machine learning algorithms, annotation was done manually. A weight score was calculated for each student using equation **1**, which was then used to classify students into their respective class. **Table 7** details the level of weighted scores and their corresponding class assignments. By using this method, 373 students were manually assigned labels corresponding to their respective weighted scores.

3.6 Data Splitting

After manual annotation, the dataset was then split into training, validation, and test sets. Original data was divided into 80% for training set and 20% for testing. The training set was further divided into 80% for training and 20% for validation.

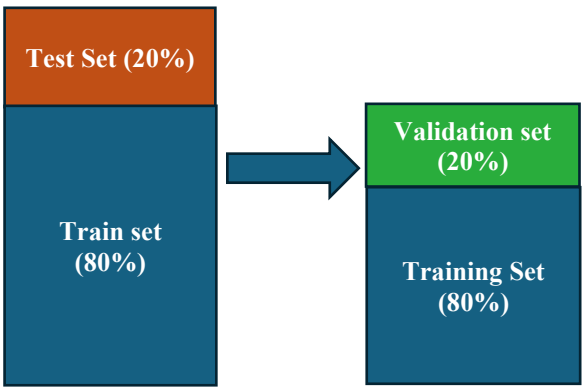

Fig. 2. Process of Data Splitting

4. Student Graduate Success Classification

In this study, to determine student success, students were categorized into four different classes: "*Failure*", "*Satisfaction*", "*Success*", and "*High Success*" using four different classifiers including Random Forests, Decision Tree, Support Vector Machine, and Logistic Regression.

4.1 Random Forests (RF)

Random forests work by aggregating a collection of numerous decision trees, each constructed randomly. Unlike a Classification and Regression Trees (CART) that seeks for optimal prediction, random forests create a "forest" of trees, some of which may not be individually optimal. With these random variations in these individual trees, it enables the forest to explore a much broader space of potential predictors, ultimately resulting in superior predictive accuracy (Genuer & Poggi, 2020). Random forests are applicable for both regression and classification tasks. For classification, the algorithm generates a class vote from each tree, with the final prediction based on the majority vote. The algorithm of random forest classification is described as follows (Hastie, Tibshirani, & Friedman, 2009):

Algorithm: Random Forest for Classification

1. For b = 1 to B:

   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size $\boldsymbol{N}$ from the training data.

   (b) Grow a random-forest tree $\boldsymbol{T_b}$ to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size $\boldsymbol{n_{min}}$ is reached.

   i. Select m variables at random from the p variables.

   ii. Pick the best variable/split-point among the m.

   iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$

To make a prediction at a new point x:

Classification: Let $\widehat{C_b}(x)$ be the class prediction of the bth random forest tree. Then $\widehat{C_{rf}^B}(x) = marjority\ vote\ \{\widehat{C_b}(x)\}_1^B$

4.2 Decision Tree (DT)

A decision tree works by breaking down classification into a series of decisions based on each feature. Commencing at the root, the process progresses down the tree to a leaf node, which provides the final classification (Marsland, 2015). In a classification tree, the quality of split is determined by an impurity measure, where a split is considered pure if all instances within each resulting branch belong to the same class. At a node m, let N_m be the number of instances that have reached this node. If N_m^i of these instances belong to class C_i (where $\sum_i N_m^i = N_m$), then the estimated probability of an instance belonging to class C_i given it reachs node m is.

$$\hat{P}(C_i|x,m) \equiv P_m^i = \frac{N_m^i}{N_m}$$

The node m is considered pure if, for all classes i, the estimated probablity P_m^i is either 0 or 1. $P_m^i = 0$ signifies no instances of class C_i reaching node m, whereas $P_m^i = 1$ shows that all instances at node m belong to class C_i. If a split results in pure nodes, no further splitting is needed, and a leaf node labeled with the class where $P_m^i = 1$ is created. There are a number of possible functions to measure impurity of the node including Gini, entropy, and log_loss (Alpaydin, 2020).

4.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a popular and powerful algorithm in modern machine learning developed by Vapnik and Chervonenkis (Marsland, 2015). The core principle of SVM is to map input vectors into a high-dimensional feature space using some non-linear functions. Within this transformed space, the algorithm constructs an optimal hyperplane that effectively separates the different classes. The optimal hyperplane is defined as the linear decision function maximizing the margin between the support vectors of the various classes. The construction of this optimal hyperplane depends solely on these support vectors, a small subset of the training data that directly determines the margin (Cortes & Vapnik, 1995). The optimal hyperplane in feature space can be written as follows (Bishop, 2007):

$$\boldsymbol{y_i(x) = w_i^T \phi(x) + b_i}$$

$\boldsymbol{\phi(x)}$: is the feature-speace mapping function

$\boldsymbol{w_i}$: is the weight vector for the i-th classifier

$\boldsymbol{b_i}$: is the bias term for the i-th classifier

SVM is fundamentally designed for binary class classification. There are a number of approaches to extend it to multiclass classification. In this study, one-versus-one has been used to classify multiclass data. In this approach, a binary SVM classifier is trained for every possible pair of classes, resulting in $\frac{K(K-1)}{2}$ binary classifiers. To predict the class for a new test point $\boldsymbol{x}$, it is passed through all these classifiers. The final class prediction for $\boldsymbol{x}$ is typically determined by a **majority voting** scheme, where the class receiving the highest number of votes is assigned (Bishop, 2007).

4.4 Logistic Regression

Similar to SVM, logistic regression is typically designed to classify only two classes. However, it can be extended to multiclass classification by using the Softmax function to transform linear functions of the feature variables. The core idea is to calculate a score for each class and then normalize these scores into probabilities using the Softmax function. For each class k, a linear score is written as follows (Bishop, 2007):

$$a_k(\phi) = w_k^T \phi$$

The Softmax function can be used to normalize these scores into probabilities:

$$P(C_k|\phi) = \frac{e^{a_k(\phi)}}{\sum_{j=1}^k e^{a_j(\phi)}}$$

Substituting $a_k(\phi) = w_k^T \phi$:

$$P(C_k|\phi) = \frac{e^{w_k^T \phi}}{\sum_{j=1}^k e^{w_j^T \phi}}$$

5. MODEL EVALUATION

In this study, the models were evaluated using five popular metrics commonly employed to assess classification models including confusion matrix, accuracy, precision, recall, and the F_1 Score.

5.1 Confusion Matrix

A confusion matrix, commonly used in machine learning to evaluate classification model performance, is an $n\ x\ n$ table where n represents the number of classes. To construct $n\ x\ n$ confusion matrices, True Positive (TP), True Negative (TN), False Negative (FN), and False Positive (FP) are calculated as follows (Marom et al., 2010, Junayed et al., 2019):

$$\boldsymbol{TP_i = a_{ii}} \tag{2}$$

$$\boldsymbol{FP_i = \sum_{j=1, j\neq i}^{n} a_{ji}} \tag{3}$$

$$\boldsymbol{FN_i = \sum_{j=1, j\neq i}^{n} a_{ij}} \tag{4}$$

$$\boldsymbol{TN_i = \sum_{j=1, j\neq i}^{n} \sum_{k=1, k\neq i}^{n} a_{jk}} \tag{5}$$

5.2 Accuracy

Accuracy, a widely used metric to assess classification model performance, is defined as the number of correct prediction and is calculated using the following equation (Junayed et al., 2019):

$$\boldsymbol{accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)}} \tag{6}$$

5.3 Precision

(7)

Precision measures the accuracy of the model in predicting positive instances. Specifically, it refers to the ratio of correctly predicted positives to the total number of instances predicted as positive. The precision is defined as follows (Junayed et al., 2019):

$$\boldsymbol{Precision = \frac{TP}{(TP + FP)}}$$

5.4 Recall

Recall measures the model's ability to identify all actual positive instances. A high recall is an indicator that the model effectively detects true positives, while a low recall suggests a high number of false negatives. The recall is defined as follows (Junayed et al., 2019):

$$\boldsymbol{Recall = \frac{TP}{(TP + FN)}} \tag{8}$$

5.5 F_1 Score

In statistical analysis of classification, F_1 Score is known as the harmonic mean of the precision and recall taking both false positive and false negatives into account. It is defined as follows (Junayed et al., 2019):

$$\boldsymbol{F_1\ Score = \frac{2 * (Recall * Precision)}{(Recall + Precision)}} \tag{9}$$

6. RESULTS AND ANALYSIS

In this study, we predict student graduate success using four machine learning algorithms: Random Forest, Decision Tree, Support Vector Machine, and Logistic Regression.

6.1 Random Forest Classifier Performance

Based on **Fig. 3**, the Random Forest Classifier (RFC) excels in predicting the "Failure" and "High Success" classes, as indicated by the absence of misclassifications of these categories. However, the "Satisfaction" and "Success" classes exhibited several False Negatives, with three instances from these actual classes being incorrectly predicted by the model. The "Success" class achieved a decent number of correct predictions (11), though 3 instances were misclassified. The Random Forest Classifier struggled significantly to predict "Satisfaction" class, making only 3 correct predictions and misclassifying 3 instances as "Failure". This shows that the model struggles to differentiate between "Satisfaction" from "Failure" class. **Table 8** summarizes the performance scores of Random Forest Classifier for each class.

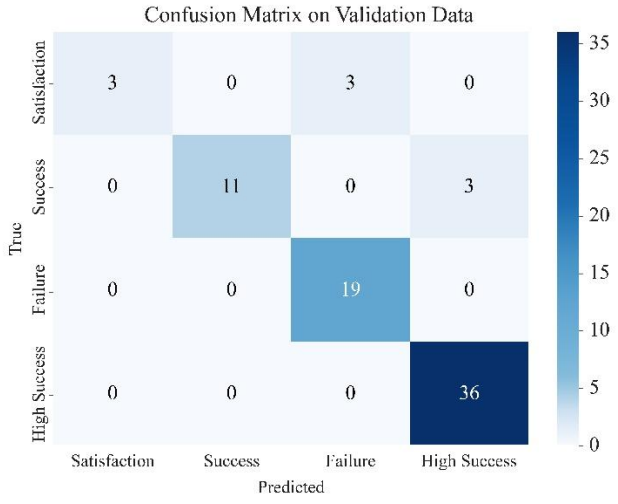

Fig. 3. Confusion Matrix: RFC on Validation Set

Table 8. Random Forest Classifier on Validation Set

Class	Precision	Recall	F1-Score
Failure	0.86	1.0	0.93
Satisfaction	1.0	0.5	0.67
Success	1.0	0.79	0.88
High Success	0.92	1.0	0.96

6.2 Decision Tree Classifier Performance

In this experiment, the Decision Tree Classifier (DTC) demonstrates strong performance for the classes: "Failure", "Satisfaction", and "High Success", as evidenced by the absence of misclassifications for these categories within the confusion matrix **Fig. 4**. This suggests perfect precision and recall for these specific classes. However, only prediction on the "Success" class exhibits False Negative. For the "Success" class, 11 instances were correctly classified. Nevertheless, 3 instances that were actually "Success" were misclassified as "High Success" class. This particular misclassification suggests that the DTC model faces difficulty in effectively distinguishing between the "Success" and "High Success" categories. The summary the performance scores of DTC for each class can be found in **Table 9**.

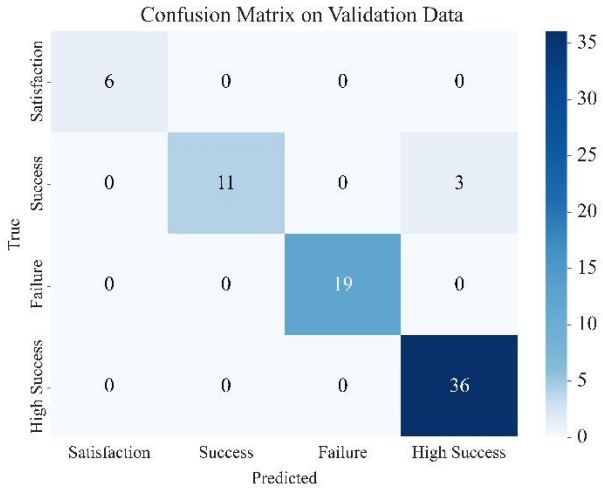

Fig. 4. Confusion Matrix: DTC on Validation Set

Table 9. Decision Tree on Validation Set

Class	Precision	Recall	F1-Score
Failure	1.0	1.0	1.0
Satisfaction	1.0	1.0	1.0
Success	1.0	0.79	0.88
High Success	0.92	1.00	0.96

6.3 Support Vector Machine Classifier Performance

According to the confusion matrix in **Fig. 5**, Support Vector Machine Classifier (SVMC) performs exceptionally well for the "High Success" category, correctly classifying 36 instances without any misclassifications. This demonstrates that the classifier is highly accurate in categorizing this specific class. However, SVMC struggles with False Negative across "Satisfaction", "Success", and "Failure" categories. While it correctly classifies 14 instances of "Failure" class, it frequently misclassifies other categories as "Failure". Specifically, 6 instances from "Satisfaction" and 1 instance from "Success" are incorrectly labeled as "Failure". Conversely, the classifier also misclassifies some "Failure" instances into other categories: 4 instances from "Failure" are incorrectly classified as "High Success", and 1 instance is misclassified as "Success." This indicates that the SVMC has difficulty in distinguishing "Failure" from the other classes.

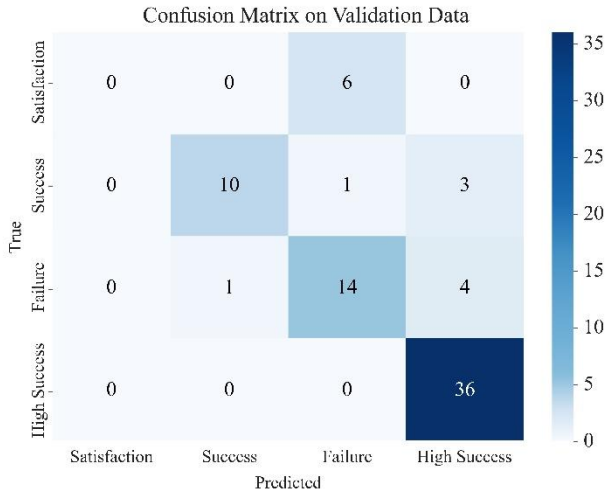

Fig. 5. Confusion Matrix: SVMC on Validation Set

Table 10. SVM Classifier on Validation Set

Class	Precision	Recall	F1-Score
Failure	0.67	0.74	0.70
Satisfaction	nan	0.00	0.00
Success	0.91	0.71	0.80
High Success	0.84	1.00	0.91

For the "Success" category, the classifier correctly identifies 10 instances. However, it makes a few misclassifications: one

instance that was actually "Failure" is incorrectly labeled as "Success." More significantly, one instance of "Success" is misclassified as "Failure," and three "Success" instances are incorrectly categorized as "High Success". SVMC performs particularly poorly for "Satisfaction" category, as it fails to correctly classify any instances. All six instances that truly belong to the "Satisfaction" category are instead misclassified as "Failure." This highlights a significant weakness in the model's ability to recognize and differentiate "Satisfaction". **Table 10** describes the detailed scores of SVMC performance on each class.

6.4 Logistic Regression Classifier Performance

Logistic Regression Classifier (LRC) excels in categorizing "High Success" class, despite a single misclassification. However, the LRC exhibits False Negative across all classes, as evident from the confusion matrix in **Fig. 6**. Specifically, the model misclassifies 4 instances of "Satisfaction" as "Failure", 2 instances of "Success" as "High Success", and 2 instances of "Failure", which are wrongly classified as "Success" and "High Success" respectively. Overall, the LRC struggles to differentiate between "Satisfaction" and "Success" classes, as well as the "Success" and "High Success" classes. **Table 11** shows the detailed scores of LRC performance for each class.

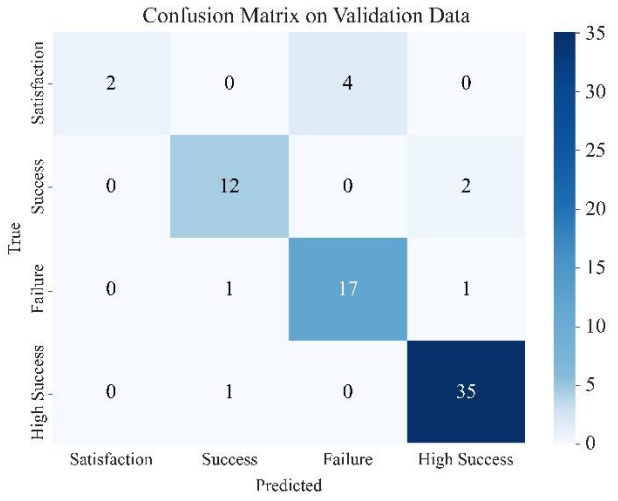

Fig. 6. Confusion Matrix: LRC on Validation Set

Table 11. Logistic Regression on Validation Set

Class	Precision	Recall	F1-Score
Failure	0.81	0.89	0.85
Satisfaction	1.00	0.33	0.50
Success	0.86	0.86	0.86
High Success	0.92	0.97	0.95

6.5 Model Performance Comparison

This comparative study investigated the efficacy of four different machine learning classifiers in predicting student graduate success, evaluating their performance on both validation and unseen test datasets. As summarized in **Table 12** and **Table *13***, the Decision Tree model consistently outperforms the other models, demonstrating exceptional accuracy across both evaluation stages. Specifically, on the validation set (**Table 12**), the Decision Tree achieved an impressive accuracy of 0.96. This was significantly higher than the other models, with Random Forest following at 0.92, Logistic Regression at 0.88, and SVM trailing at 0.80. All the classifiers enhanced slightly on the test set, as shown in **Table 13**, suggesting a more robust indicator of real-world generalization. The Decision Tree model improved its accuracy to an outstanding 0.98, solidifying its superior predictive capability. Similarly, all other classifiers also exhibited minor improvements on the test set, suggesting robust model performance without signs of significant overfitting. Random Forest's accuracy rose to 0.93, Logistic Regression to 0.89, and SVM to 0.84. The results strongly suggest that tree-based models (Decision Tree and Random Forest) are exceptionally well-suited for categorizing student graduate success within this problem domain and dataset. Their hierarchical decision-making process appears to effectively capture the underlying patterns in the data. Conversely, the SVM model consistently yielded the lowest accuracy among all classifiers, indicating that its approach was less effective for this specific problem and dataset compared to the other evaluated methods.

Table 12. Overall Accuracy on Validation Set

Classifier	Accuracy
Random Forest	0.92
Decision Tree	0.96
Logistic Regression	0.88
SVM	0.8

Table 13. Overall Accuracy on Test Set

Classifier	Accuracy
Random Forest	0.93
Decision Tree	0.98
Logistic Regression	0.89

SVM	0.84

This comprehensive study reveals a counterintuitive finding: Decision Tree emerges as the most promising model for accurately predicting student graduate success. This stands in direct contrast to established machine learning theory, which generally asserts that ensemble methods like Random Forest should outperform individual decision trees due to their ability to reduce overfitting and improve generalization through the aggregation of multiple trees. Further analysis necessitates to validate the specific characteristics of the student success prediction problem.

6.6 K-Fold Cross Validation

With this unexpected outcome, we performed a deeper examination to validate these results and explore the underlying reasons for Decision Tree's unexpected superiority by conducting k-fold cross validation. This technique is essential for robust model evaluation, as it assists in assessing how well our model generalizes to unseen data and mitigates the risk of overfitting. In this experiment, we employed 5-fold cross validation by setting the parameter k to 5, meaning that we divided our dataset into five equally sized folds. The selection of k=5 is a common practice in machine learning and often effective choice, striking a balance between computational cost and bias-variance trade-off (James et al., 2013). Both Tree-based classifiers (Random Forest and Decision Tree) show excellent performance with very high mean scores (0.963 and 0.966, respectively) and low standard deviations (0.022, 0.024, respectively), indicating strong and consistent predictive capability. Logistic Regression and SVM performed considerably worse, suggesting they are less suitable for this particular classification task and dataset compared to three-based methods.

Table 14. 5-Fold Cross Validation Results

Classifier	Mean	Standard Deviation
Random Forest	0.963	0.022
Decision Tree	0.966	0.024
Logistic Regression	0.885	0.030
SVM	0.785	0.038

As shown in **Table 14**, the Decision Tree slightly outperforms Random Forest on average. As noted in our result and analysis, this is an interesting observation as ensemble methods like Random Forest typically perform better than individual decision trees. However, Decision Tree has a higher standard deviation suggests that the Decision Tree classifier's performance varied a bit more across the various folds of the cross validation. Decision Tree achieves a marginally higher mean performance (0.966) but is slightly less stable with higher standard deviation 0f 0.024, whereas Random Roest achieves a very slight mean performance (0.963) but is marginally more stable with lower standard deviation of 0.022. Initial 5-fold cross validation revealed that the mean and standard deviation of Random Forest and Decision Tree were not significantly different. To further validate these observations and obtain a more robust assessment, we subsequently conducted 10-fold cross validation. The results of this extended analysis are summarized in **Table 15**.

Table 15. 10-Fold Cross Validation Results

Classifier	Mean	Standard Deviation
Random Forest	0.956	0.029
Decision Tree	0.962	0.038
Logistic Regression	0.885	0.034
SVM	0.805	0.039

As evident in **Table 15**, Decision Tree received a much higher standard deviation compared to Random Forest, implying that the Decision Tree's **performance** is less consistent and more variable when trained and tested on different subsets of the data. The lower standard deviation of the Random Forest indicates greater robustness and stability.

7. CONCLUSIONS

In conclusion, while Decision Tree exhibits a slightly higher mean than Random Forest, it also possesses a significantly higher standard deviation. This considerably higher standard deviation implies that Decision Tree's performance is less consistent and potentially less reliable in real-world scenarios of student graduate success classification. When deploying these models and continuously collecting new data to retrain them, the Decision Tree might demonstrate slightly more fluctuation in its predictive accuracy or other performance metrics over time, depending on the specific new data it encounters for training. In contrast, Random Forest, being an ensemble of many trees, tends to average out the individual tree's biases and variances, leading to a more consistent overall performance. Consequently, the Random Forest model becomes more stable and robust. The experimental results show that our proposed model enables Higher Education Institutions to assess the level of success of their students after graduation. The classification results might be able to improve with deep learning model when huge amounts of data are available.

REFERENCES

[1] Ayán, M. N. R., & García, M. T. C. (2008). Prediction of university students' academic achievement by linear and logistic models. The Spanish journal of psychology, 11(1), 275-288.

[2] Essa, A., & Ayad, H. (2012). Improving student success using predictive models and data visualisations. *Research in learning technology, 20*.

[3] Jayaprakash, S., Balamurugan, E., & Chandar, V. (2015). Predicting Students' Academic Performance Using Naïve Bayes Algorithm. In *8th Annual International Applied Research Conference*.

[4] Belachew, E. B., & Gobena, F. A. (2017). Student performance prediction model using machine learning approach: the case of Wolkite university. *International Journal of Advanced Research in Computer Science and Software Engineering*, 7(2), 46-50.

[5] Oyerinde, O. D., & Chia, P. A. (2017). Predicting students' academic performances–A learning analytics approach using multiple linear regression. *International Journal of Computer Applications*, *157*(4), 37-44.

[6] Oloruntoba, S. A., & Akinode, J. L. (2017). Student academic performance prediction using support vector machine. *International Journal of Engineering Sciences and Research Technology*, 6(12), 588-597.

[7] Gerritsen, L., & Conijn, R. (2017). Predicting student performance with Neural Networks. *Tilburg University, Netherlands*.

[8] Pojon, M. (2017). *Using machine learning to predict student performance* (Master's thesis, University of Tampere). Trepo. https://trepo.tuni.fi/handle/10024/101646.

[9] Ulinnuha, N., Sa'Dyah, H., & Rahardjo, M. (2017). *A Study of Academic Performance Using Random Forest, Artificial Neural Network, Naïve Bayesian and Logistic Regression*.

[10] Obsie, E. Y., & Adem, S. A. (2018). Prediction of student academic performance using neural network, linear regression and support vector regression: a case study. *International Journal of Computer Applications*, 180(40), 39-47.

[11] Kim, B. H., Vizitei, E., & Ganapathi, V. (2018). Gritnet 2: Real-time student performance prediction with domain adaptation. *arXiv preprint arXiv:1809.06686*, 1-8.

[12] Buenaño-Fernández, D., Gil, D., & Luján-Mora, S. (2019). Application of machine learning in predicting performance for computer engineering students: A case study. *Sustainability, 11*(10), 2833.

[13] Rajalaxmi, R. R., Natesan, P., Krishnamoorthy, N., & Ponni, S. (2019). Regression model for predicting engineering students academic performance. *International Journal of Recent Technology and Engineering*, *7*(6S3), 71-75.

[14] Sekeroglu, B., Dimililer, K., & Tuncal, K. (2019, March). Student performance prediction and classification using machine learning algorithms. In Proceedings of the 2019 8th international conference on educational and information technology (pp. 7-11).

[15] Zohair, A., & Mahmoud, L. (2019). Prediction of Student's performance by modelling small dataset size. *International Journal of Educational Technology in Higher Education*, *16*(1), 1-18.

[16] Lau, E. T., Sun, L., & Yang, Q. (2019). Modelling, prediction and classification of student academic performance using artificial neural networks. *SN Applied Sciences*, *1*(9), 982.

[17] Anderson, H., Boodhwani, A., & Baker, R. (2019, March). Predicting graduation at a public R1 university. In *Proceedings of the 9th International Learning Analytics and Knowledge Conference*.

[18] Baashar, Y., Alkawsi, G., Mustafa, A., Alkahtani, A. A., Alsariera, Y. A., Ali, A. Q., ... & Tiong, S. K. (2022). Toward predicting student's academic performance using artificial neural networks (ANNs). *Applied Sciences, 12*(3), 1289.

[19] Genuer, R., & Poggi, J. M. (2020). Random Forests with R. Springer Nature.

[20] Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: springer.

[21] Marsland, S. (2015). Machine Learning: An Algorithmic Perspective. CRC Press.

[22] Alpaydin, E. (2020). Introduction to Machine Learning. MIT Press.

[23] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20, 273-297.

[24] Bishop, C. M. (2007). Pattern Recognition and Machine Learning. Switzerland: Springer.

[25] Marom, N. D., Rokach, L., & Shmilovici, A. (2010, November). Using the confusion matrix for improving ensemble classifiers. In 2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel (pp. 000555-000559). IEEE.

[26] Junayed, M. S., Jeny, A. A., Atik, S. T., Neehal, N., Karim, A., Azam, S., & Shanmugam, B. (2019, July). AcneNet-a deep CNN based classification approach for acne classes. In 2019 12th International Conference on Information & Communication Technology and System (ICTS) (pp. 203-208). IEEE.

[27] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning: With applications in R. Springer.