TIERS Information Technology Journal
Vol.6, No.1, June 2025, pp. 129~143
P-ISSN: 2723-4533 / E-ISSN: 2723-4541
DOI: https://doi.org/10.38043/tiers.v6i1.6806

Machine Learning-Based Classification of Student Adaptability in Online Learning with Feature Engineering

Yasin Efendi
yasin.efendi@umj.ac.id
Information Technology Education Study Program, Faculty of Education, Universitas Muhammadiyah Jakarta, Banten, Indonesia

ABSTRACT

Student adaptability in online learning environments has become increasingly important in contemporary education. This study introduces a feature engineering approach guided by SHAP (SHapley Additive exPlanations) to enhance the classification of student adaptability levels. Unlike prior studies that primarily utilize exploratory analysis or statistical importance scores, this method leverages SHAP values to construct new features by considering both statistical contribution and semantic meaning. Three additional features were created by combining original variables, representing educational level and session duration, digital access quality, and socioeconomic context. The dataset was evaluated using four classic machine learning models, namely Support Vector Machine (SVM), k-Nearest Neighbors (KNN), Decision Tree, and Random Forest, both before and after applying the engineered features. Results show that SHAP-based feature engineering improved model performance in most cases. The most notable gains were observed in Decision Tree and Random Forest models, where the F1-score increased from 84.87% to 89.34% and from 85.80% to 89.34%, respectively, while accuracy rose from 88.38% to 90.08% and from 89.63% to 90.08%, respectively. The SVM model also recorded an increase in recall from 82.49% to 87.28%, whereas KNN showed a slight drop in accuracy but improved in ROC AUC from 91.55% to 93.83%. These findings demonstrate that explainable feature design not only enhances accuracy and F1-score, particularly in tree-based models, but also supports model interpretability, enabling more transparent, reliable, and effective educational decision-making systems.

Keywords: Feature Engineering; Machine Learning; Online Learning; SHAP; Student Adaptability

Article Info

Received : 01-03-2025
Revised : 21-04-2025
Accepted : 30-06-2025

Correspondence Author:

Yasin Efendi
Information Technology Education Study Program, Faculty of Education,
Universitas Muhammadiyah Jakarta,
Jl. KH. Ahmad Dahlan, Cirendeu, Ciputat, Tangerang Selatan. Banten, Indonesia
Email: yasin.efendi@umj.ac.id

1. INTRODUCTION

Online learning has become a central pillar of modern education systems, especially since the global pandemic compelled educational institutions to shift to remote learning models. In this context, students' ability to adapt to digital learning environments has become critically important. Student adaptability reflects the extent to which learners can adjust cognitively, affectively, and behaviorally to the demands of online learning [1]. This adaptability affects not only learning engagement and motivation but also directly contributes to academic achievement [2]. Therefore, automatically classifying student adaptability levels using machine learning is highly relevant for designing adaptive and personalized learning strategies.

Several prior studies have attempted to classify students' adaptability levels using machine learning approaches. One study employed the Random Forest algorithm with a One-vs-Rest strategy, involving categorical variable encoding and class imbalance handling using SMOTE [3]. The model achieved an accuracy of 88.38%, yet it did not assess the significance of the features used in the classification process. Another study utilized GNB (Gaussian Naive Bayes) and KNN (*k*-Nearest Neighbors), with parameter optimization through GridSearchCV [4]. The ROC-AUC score of KNN reached 0.89, outperforming GNB at 0.81. However, the study did not explicitly address class imbalance or analyze feature contributions to model performance. A different work compared models such as KNN, Decision Tree, Random Forest, Naïve Bayes, SVM (Support Vector Machine), and ANN (Artificial Neural Network) [5]. Although Random Forest yielded a high accuracy of 89.63%, the study did not incorporate strategies tailored to imbalanced data handling. Furthermore, other research contrasted various machine learning algorithms such as Random Forest, SVM, Logistic Regression, KNN, and Extreme Gradient Boosting (XGBoost), and reported that both Random Forest and XGBoost achieved the top accuracy of 92% [6]. Although feature importance techniques were employed to identify relevant variables, there was no feature combination or engineering process to enrich the representation of existing features.

Based on previous studies, one of the main challenges in classifying students' adaptability levels in online learning environments is class imbalance. In real-world educational datasets, the distribution of students across different adaptability levels (such as Low, Medium, and High) is often uneven. This imbalance can lead to biased predictions, where the majority class dominates the model's learning process, reducing its ability to perform well in terms of recall and F1-score for the minority classes [7]. Addressing class imbalance is crucial for achieving fairness, accuracy, and robustness in predictive performance across various classroom conditions. One widely used strategy to handle this issue is oversampling the minority class, such as through the Synthetic Minority Over-sampling Technique (SMOTE), which generates synthetic samples based on nearest neighbors [8]. SMOTE helps balance class distribution without removing existing data, allowing the model to learn from each adaptability category more effectively.

However, while class imbalance is a critical issue, another equally important challenge lies in the feature engineering process. Many models still rely on raw variables without deeper analysis of their relative importance or the interactions between features. In machine learning, the selection of the appropriate algorithm alone is not enough; how features are represented and constructed plays an equally vital role in model success. Feature engineering refers to the process of creating new features or transforming existing ones to better capture meaningful patterns in the data [9]. Some studies have utilized feature engineering to improve machine learning model performance. For instance, a study on predicting students' academic performance showed that applying feature engineering increased average accuracy by 0.33% and ROC AUC by 0.63% [10]. However, the feature engineering techniques used were still exploratory and lacked a systematic methodological framework. Other studies have explored SHAP-based (SHapley Additive exPlanations) feature engineering and importance-based methods in classification tasks such as credit card fraud detection, yielding modest improvements in model performance [11]. In another study, feature engineering was applied to enhance machine learning in identifying key factors affecting learning outcomes, where Support Vector Regression (SVR) with engineered features provided the most accurate predictions [12].

In response to gaps in earlier studies, this work presents a SHAP-driven feature engineering method to boost classification accuracy of student adaptability in online learning. SHAP was chosen for its high level of interpretability, providing both detailed and overarching understanding of how individual features influence the model's predictions [13]. This allows for the creation of new features that are not only intuitive but also statistically guided. To assess the impact of this feature engineering process, four classical machine learning algorithms were employed, namely SVM, KNN, Decision Tree, and Random Forest. These models were chosen to represent diverse learning paradigms: margin-based, distance-based, tree-based, and ensemble learning, thereby offering a comprehensive comparative evaluation [14].

The primary objective of this study is to enhance classification accuracy for student adaptability by creating new, SHAP-informed features that capture important interactions between variables, such as education level, class duration, and students' socioeconomic conditions. By focusing on SHAP-based feature engineering, this study aims to improve model performance, especially for tree-based algorithms, compared to the baseline model with original features. The hypothesis of this study is that SHAP-based feature engineering will significantly improve the classification performance of machine learning models, particularly for tree-based algorithms, in comparison to using the original features alone. The novelty of this work lies in the generation of new features from SHAP values and the comprehensive comparison of the models' predictive performance before and after applying feature transformations, measured by accuracy, precision, recall, F1-score, and ROC AUC.

2. RESEARCH METHOD

This study aims to improve the classification of students' adaptability levels in online learning through a SHAP-based (SHapley Additive exPlanations) feature engineering technique. The evaluation utilized four

.

traditional machine learning algorithms: SVM, KNN, Decision Tree, and Random Forest. The research methodology is depicted through the procedural diagram in Figure 1.

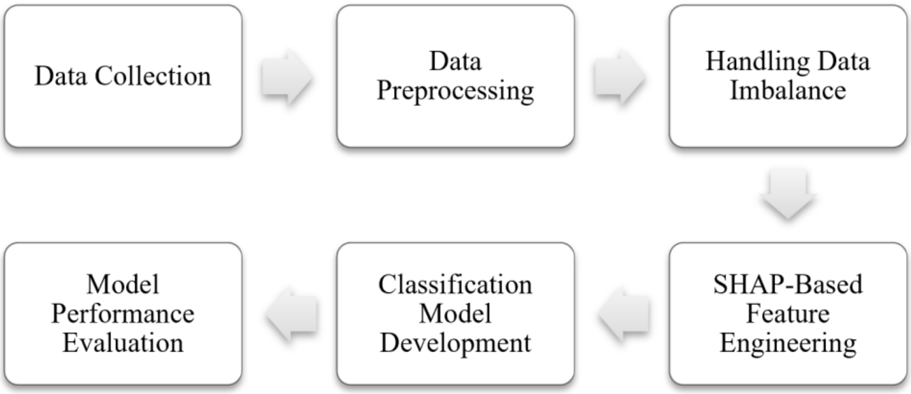


Figure 1. Diagram illustrating the research workflow

A detailed explanation of each step illustrated in Figure 1 is provided in the following subsections.

2.1. Data Collection

The dataset used in this study was sourced from the Kaggle platform, titled *Students Adaptability Level in Online Education* [15]. It contains 1,200 entries, each representing a student and includes 13 predictor variables and one target variable. The predictor variables encompass demographic and operational factors, such as gender, age, education level, institution type, IT student status, geographic location, load-shedding status, family financial condition, internet type, network type, online class duration, use of Learning Management System (LMS), and the type of device used. The target variable, Adaptivity Level, is categorized into three classes: Low, Moderate, and High. The distribution of labels is imbalanced, with the "Moderate" and "High" classes being overrepresented, while the "Low" class is underrepresented, creating a class imbalance issue that requires attention during model training. Table 1 provides an overview of the dataset, detailing the types of features and their descriptions.

Table 1. Dataset Characteristics

Feature	Type	Description
Gender	Categorical	Gender of the student
Age	Categorical	Age range of the student
Education Level	Categorical	Education level (e.g., University, College, School)
Institution Type	Categorical	Type of institution (Government, Non-Government)
IT Student	Categorical	Whether the student is an IT student (Yes/No)
Location	Categorical	Geographical location (e.g., Urban, Rural)
Load-shedding	Categorical	Load-shedding status (e.g., Low, High)
Financial Condition	Categorical	Student's financial condition (e.g., Poor, Mid, High)
Internet Type	Categorical	Type of internet connection (e.g., Wifi, Mobile Data)
Network Type	Categorical	Network type (e.g., 4G, 3G)
Class Duration	Categorical	Duration of the online class (e.g., 0, 1-3, 3-6 hours)
Self LMS	Categorical	Whether the student uses LMS (Yes/No)
Device	Categorical	Device used for online learning (e.g., Mobile, Laptop, Tablet)
Adaptivity Level	Categorical	Target variable: Adaptability level (e.g., Low, Moderate, High)

As shown in Table 1, the dataset consists of categorical variables that describe various student characteristics. The target variable, Adaptivity Level, is crucial for assessing the students' adaptability in online learning environments, which is influenced by various demographic and operational factors.

2.2. Data Preprocessing

The preprocessing stage began with transforming all categorical features into numerical format using Label Encoding to enable processing by numerical-based classification algorithms. All numerical features were then normalized using *StandardScaler* to standardize feature scales and prevent dominant influence due to value ranges. After normalization, the dataset was split into training and testing subsets with an 80:20 ratio using stratified sampling. This stratification ensures that the class distribution is preserved across both subsets, maintaining representativeness for the entire population [16].

2.3. Handling Data Imbalance

SMOTE (Synthetic Minority Oversampling Technique) was utilized to handle the issue of imbalanced data distribution. SMOTE is a widely used method to enhance the distribution of minority classes in classification tasks [17], as imbalance often leads to model bias toward the majority class, which can result in misleading accuracy and poor minority class performance [18].

SMOTE creates additional synthetic instances by performing linear interpolation between a selected minority instance x_i and one of its nearest neighbors x_{zi}, as described in Equation (1).

$$x_{new} = x_i + \delta \times (x_{zi} - x_i) \tag{1}$$

where x_i is the feature vector of a minority class sample, x_{zi} is the feature vector of one of its nearest neighbors, and δ represents a randomly chosen value within the range of 0 to 1. This formula generates a new data point x_{new} positioned between two instances of the minority class in the feature space, effectively enhancing the minority class distribution without repeating existing samples.

SMOTE helps form a denser and more representative feature space for the minority class, enabling classification algorithms to establish a more balanced decision boundary [8]. In this study, SMOTE was applied only to the training set, while the test set retained its original class distribution. This method maintains the integrity of model assessment by simulating its actual performance on skewed datasets. It seeks to enhance the model's ability to detect the minority class while retaining high overall accuracy.

2.4. SHAP-Based Feature Engineering

Improving data representation through feature engineering is a key component of the machine learning process, enabling models to better capture relevant patterns [19]. This research employed a SHAP-based (SHapley Additive exPlanations) method to perform feature engineering. SHAP is a model interpretation technique that quantifies the additive contribution of each feature to the prediction output [20]. SHAP (SHapley Additive exPlanations) has been increasingly adopted in educational research due to its ability to enhance model interpretability and support data-driven decision-making. One study applied SHAP to identify key features influencing students' adaptability, enabling more targeted educational interventions [21]. Another study utilized SHAP to analyze the factors affecting students' active transportation behaviors, offering insights to improve school accessibility and public health in underserved regions [22]. Additionally, SHAP has been employed to improve early detection of dropout risks among undergraduate students, demonstrating its effectiveness in identifying influential predictors [23]. Other research highlights the critical importance of interpretability in education, positioning SHAP as a valuable tool to ensure transparency in the application of machine learning models [24].

The process began by training an initial Random Forest model on the SMOTE-enhanced training data to compute SHAP values for each feature with respect to the classification output. Random Forest was selected due to its decision-tree ensemble structure, natural interpretability, and effectiveness in handling both numerical and categorical data [25]. Furthermore, Random Forest is non-parametric and relatively robust to outliers, allowing for more reliable and representative estimations of feature contributions for SHAP calculation. SHAP values were computed across all features to obtain their mean absolute contribution to the model prediction, calculated using Equation (2).

$$Mean\ Absolute\ SHAP\ Value_J = \frac{1}{N}\sum_{i=1}^{N}\left|\varphi_{i,j}\right| \tag{2}$$

where $\varphi_{i,j}$ ais the SHAP value for feature j on sample i, N a is the total number of samples, and j denotes the feature index. Absolute values prevent opposing contributions from canceling out.

Features with the highest SHAP values were then selected as the basis for generating new features. Feature construction was performed using various approaches, including combining complementary features, applying logarithmic transformation to features with skewed distributions to stabilize variance, and creating binary features based on specific SHAP-derived thresholds [26]. The SHAP-based feature engineering process, as illustrated in Figure 2.

.

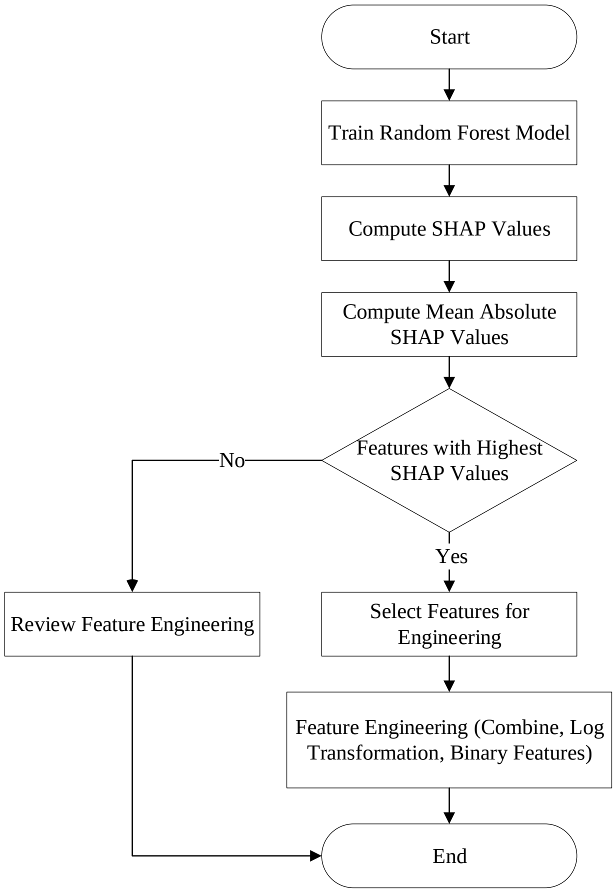


Figure 2. SHAP-based Feature Engineering Process for Classifying Student Adaptability

The flowchart in Figure 2 visually summarizes the entire SHAP-based feature engineering process. It highlights the key steps, from training the initial Random Forest model to compute SHAP values, calculating the mean absolute SHAP values, and finally selecting the features with the highest SHAP values for engineering. This process ensures that the generated features are not only statistically justified but also semantically relevant, improving the interpretability and accuracy of the classification models. Through this process, feature engineering was guided not merely by exploratory analysis but by strong statistical justification, both locally (individual data points) and globally (entire dataset).

2.5. Classification Model Development

This study employed four widely used classical machine learning algorithms for classification tasks due to their efficiency, interpretability, and scalability, particularly in educational contexts where computational constraints and transparency are essential [27]. The study employed SVM, KNN, Decision Tree, and Random Forest to represent margin-based, distance-based, tree-based, and ensemble learning paradigms.

The SVM algorithm classifies data by finding a hyperplane that maximally separates two classes in the feature space [28]. It is especially well-suited for handling binary classification problems involving high-dimensional feature spaces. The decision function of SVM is defined by Equation (3).

$$f(x) = \text{sign}(w^T x + b) \tag{3}$$

where w represents the weight vector, x denotes the input features, and b is the bias. refers to the bias term. The ideal hyperplane is found by expanding the margin between classes as much as possible while simultaneously reducing classification errors.

KNN (k-Nearest Neighbors) is a non-parametric algorithm that classifies a data point based on the most common class among its k closest training samples in the feature space [29]. Class prediction is based on the Euclidean distance to all training samples, and the majority class among the closest k samples is selected. The Euclidean distance between x_i and x_j is defined by Equation (4).

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^{n} (x_{il} - x_{jl})^2} \tag{4}$$

where n is the number of features, and x_{il}, x_{jl} are the values of the l^{th} feature for x_i and x_j, respectively.

A Decision Tree is a classification algorithm that organizes data by recursively partitioning it according to input feature values, resulting in a tree-like decision structure [30]. Feature selection at each node is based on impurity measures such as the Gini Index or Entropy. In this study, the Gini Index was used, defined by Equation (5).

$$Gini = 1 - \sum_{i=1}^{C} p_i^{\,2} \tag{5}$$

where C is the number of classes and p_i is the relative probability of class i at a node. Lower Gini values indicate higher node homogeneity.

Random Forest is an ensemble-based method that generates multiple decision trees by randomly sampling both data instances and feature subsets. It aggregates the predictions from all trees through majority voting to determine the final outcome [31]. Random Forest is known for reducing overfitting and improving accuracy. The final prediction of Random Forest is given by Equation (6).

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \cdots, h_T(x)\} \tag{6}$$

where $h_t(x)$ is the prediction from the t^{th}, and T is the total number of trees in the ensemble.

All four models were tested in two distinct scenarios, namely before and after SHAP-based feature engineering, to objectively evaluate the impact of feature transformation on the performance of each algorithm. The key hyperparameters used for training each model in their final configuration are presented in Table 2.

Table 2. Final hyperparameter settings for each classification model

Model	Key Hyperparameters
SVM	Kernel = RBF, C = 1.0, Gamma = scale
KNN	n_neighbors = 5, metric = Euclidean
Decision Tree	Criterion = Gini, max_depth = None, min_samples_split = 2
Random Forest	n_estimators = 100, Criterion = Gini, max_depth = None, min_samples_split = 2

Table 2 presents the hyperparameters used for each model. These parameters were selected through a combination of grid search and empirical tuning to achieve optimal performance in both the baseline and SHAP-based feature engineering scenarios.

2.6. Model Performance Evaluation

Model performance was evaluated using a combination of quantitative metrics and ROC curve visualization to provide a comprehensive assessment. The evaluation involved the four classification algorithms both before and after SHAP-based feature engineering. The five primary metrics used included accuracy, precision, recall, F1-score, and ROC AUC. Precision, recall, and F1-score were calculated using macro averaging to accommodate the class imbalance. In addition, Receiver Operating Characteristic (ROC) curves were visualized for each model using a One-vs-Rest scheme to evaluate their ability to distinguish between classes [32]. The Area Under the Curve (AUC) was used as an indicator of each model's discriminative performance [33]. The evaluation results from both scenarios were compared to assess the extent to which feature engineering improved the model's effectiveness and generalizability in classifying students' adaptability levels in online learning environments.

3. RESULTS AND DISCUSSION

3.1. Data Preparation and Class Balance

The dataset used in this study was obtained from the public Kaggle platform titled *Students Adaptability Level in Online Education* [15]. It consists of 1,200 entries, where each entry represents a student and includes 13 predictor attributes and one target attribute indicating the student's adaptability level to online learning. Initial preprocessing was carried out to ensure the data format was compatible with machine learning algorithms. All categorical features were converted to numerical form using label encoding, enabling compatibility with classification models that require numerical input. Subsequently, numerical features were standardized using the *StandardScaler* to align feature ranges and prevent any single attribute from dominating due to scale differences. After preprocessing, the target label proportions were analyzed to assess class distribution. The initial class distribution visualization is presented in Figure 3.

.

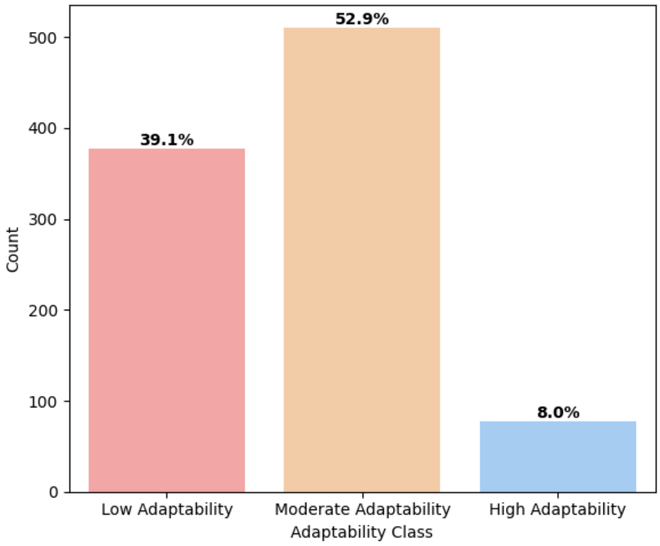


Figure 3. Initial distribution of adaptivity level classes

Figure 3 illustrates a clear class imbalance. The Moderate Adaptability class dominates with 52.9% of the data, followed by Low Adaptability with 39.1%, and only 8.0% of entries belonging to the High Adaptability class. This imbalance may cause the classification model to bias toward the majority class and fail to make accurate predictions for minority classes. To mitigate this problem, SMOTE was implemented using the *imblearn* library within the Google Colab platform. This oversampling process was performed only on the training data, which had been separated beforehand, to avoid data leakage. The new class distribution after SMOTE application is depicted in Figure 4.

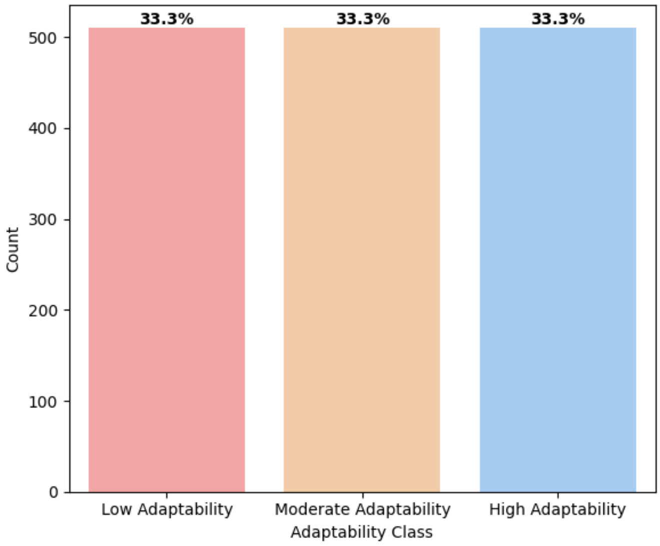


Figure 4. Class distribution after SMOTE oversampling

Figure 4 shows that the three classes now have an equal share, approximately 33.3% each within the training set. This balanced condition ensures that the classification models can learn from all classes fairly and reduces the risk of bias toward the majority class.

3.2. SHAP-Based Feature Importance Analysis

To assess the contribution of each feature to model predictions, this study utilized SHAP (SHapley Additive exPlanations), an interpretability method derived from cooperative game theory that treats features as contributors to the model output. SHAP calculates each feature's marginal contribution by evaluating all possible feature combinations, offering both fair and consistent local-global explanations.

In this study, SHAP was implemented by first training a Random Forest model on the SMOTE-balanced training data. Random Forest was selected because it is a tree-based ensemble model that naturally supports interpretability, handles both numerical and categorical data effectively, and is robust against outliers. These characteristics make it reliable for producing representative estimations of feature contributions. Subsequently, SHAP values were computed for each feature in relation to predictions of the three adaptability classes: Low, Moderate, and High. The aggregated SHAP values are visualized in Figure 5.

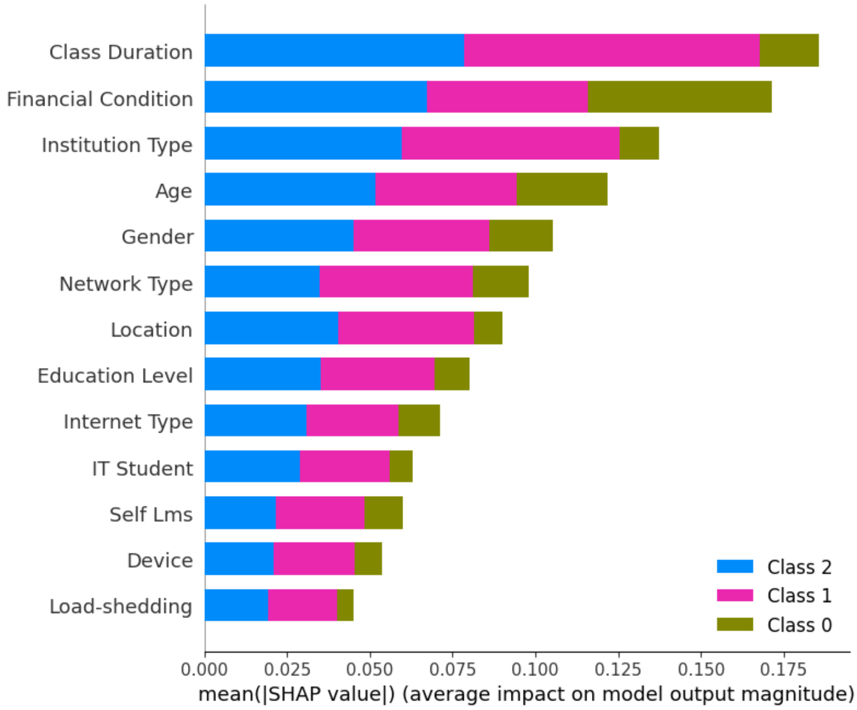


Figure 5. Visualization means the absolute SHAP value of each feature against the target class

Figure 5 illustrates the average absolute contribution of each feature to the model's output, segmented by target class. Blue, magenta, and olive green represent contributions toward predicting class 2 (High), class 1 (Moderate), and class 0 (Low), respectively. The analysis revealed that the features Class Duration, Financial Condition, and Institution Type have the most significant impact on model predictions across all classes. Class Duration consistently shows high contribution to all adaptability levels, indicating that the amount of time spent in online classes is closely related to students' adaptability. Similarly, Financial Condition and Institution Type exert substantial influence, highlighting the role of economic background and institution type in supporting online learning engagement. On the other hand, features such as Load-shedding, Device, and Self LMS exhibit relatively lower contributions, suggesting that within this dataset, these factors are less influential in distinguishing students' adaptability levels. These SHAP insights serve as the foundation for the subsequent feature engineering process, where features with high SHAP values are considered for transformation, combination, or further refinement to enhance the model's discriminative capacity.

3.3. Feature Engineering Based on SHAP

Following the SHAP-based analysis of feature importance, the next step involved designing new features that capture potential interactions among existing attributes. The aim was to enrich the feature space with semantically meaningful combinations that could improve model discrimination. The engineered features were created by combining original attributes that were both statistically relevant (based on SHAP values) and contextually aligned with online learning adaptation.

The selection of features for combination was guided by three interconnected considerations. First, only attributes that consistently ranked high in the SHAP importance list across multiple models were chosen to ensure strong predictive relevance. Second, the attributes needed to demonstrate a potential interaction effect, meaning that when combined, they could form a meaningful construct in the context of online learning adaptability, such as merging academic engagement indicators with measures of resource availability. Finally, the combinations were required to align with domain knowledge, ensuring that they reflected established educational factors known to influence students' ability to adapt. Table 3 presents the newly constructed features along with their logical justifications.

.

Table 3. Engineered features and justification

New Feature	Original Features	Justification
Edu_Duration	Education Level × Class Duration	Represents the relationship between education level and learning habits in online education. Students at higher education levels tend to have longer class durations, which may influence their adaptability.
Device_Internet	Device × Internet Type	Describes the quality of access to online learning. Combinations such as computer + Wi-Fi provide a more stable learning experience compared to mobile phones + mobile data.
Location_Financial	Location × Financial Condition	Represents the socioeconomic dimension of students. Those from rural areas with lower economic conditions may face greater challenges in adapting compared to urban students with good financial conditions.

Table 3 presents the new features generated through the feature engineering process based on SHAP analysis. Each feature represents a combination of two original attributes that are conceptually related and believed to capture important aspects of the online learning adaptability process. The *Edu_Duration* feature reflects the influence of education level on students' engagement intensity in online classes. The *Device_Internet* feature combines the dimensions of device and internet connection as an indicator of learning access quality. Meanwhile, *Location_Financial* represents the students' socioeconomic conditions, which may affect their ability to adapt to online education systems.

After combining features based on the SHAP value analysis, an evaluation was conducted to assess the usefulness of the newly constructed attributes using a correlation heatmap. This heatmap was used to examine the relationships among both original and engineered features to ensure that the new features were not redundant and provided additional, relevant information for predicting the target variable. The correlation results indicate that *Edu_Duration* has a strong correlation with both *Education Level* (0.73) and *Class Duration* (0.71), suggesting that it effectively captures the joint information of its components. *Device_Internet* is highly correlated with *Internet Type* (0.74) and moderately with *Device* (0.43), reflecting its ability to represent digital access quality. Meanwhile, *Location_Financial* shows a very high correlation with *Financial Condition* (0.91) and a fair correlation with *Location* (0.26), indicating its validity in representing students' socio-economic conditions. Overall, these engineered features do not introduce excessive multicollinearity and contribute valuable information to the predictive model. The correlation patterns are illustrated in Figure 6.

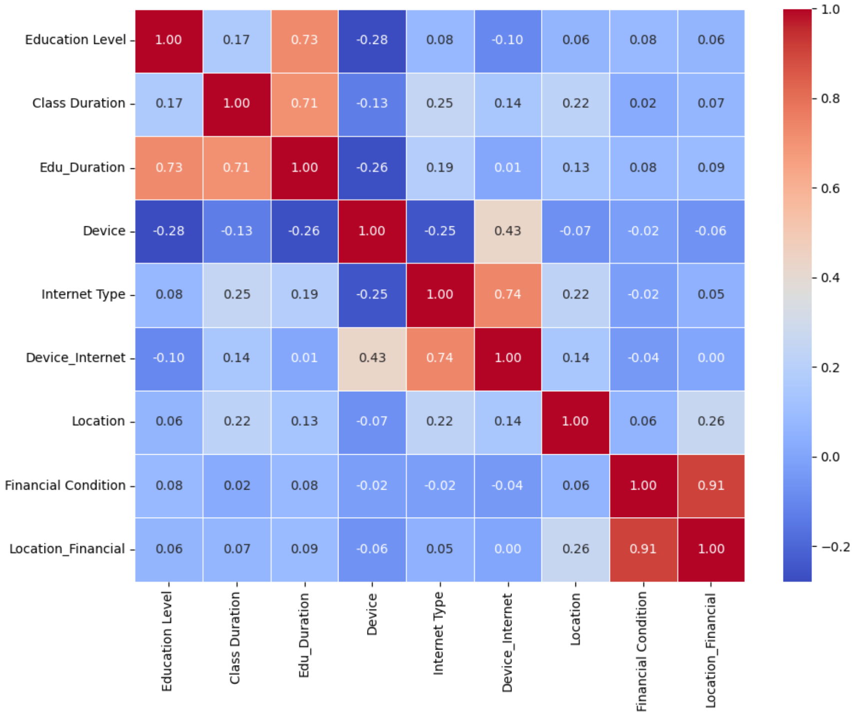


Figure 6. Correlation heatmap of original and engineered features

3.4. Model Building and Performance Evaluation

The model implementation was conducted in the Google Colab environment, which provides cloud-based computational resources and supports various machine learning libraries. Using Scikit-learn's *train_test_split*, the data was partitioned into 80% for training and 20% for testing purposes. A fixed *random_state* parameter was applied to ensure result reproducibility. Additionally, the class distribution in the target variable was preserved during the split to ensure a representative evaluation of the model's performance. Four classical machine learning algorithms were used in this study: SVM, KNN, Decision Tree, and Random Forest. SVM constructs an optimal hyperplane to separate classes, KNN classifies samples based on the majority of their nearest neighbors, Decision Tree builds classification rules through decision tree structures, while Random Forest aggregates multiple decision trees to improve accuracy and reduce overfitting. All model implementations were performed using Scikit-learn in Google Colab, with model training conducted via the *.fit()* method and evaluation through *.predict()* and *.predict_proba()* functions.

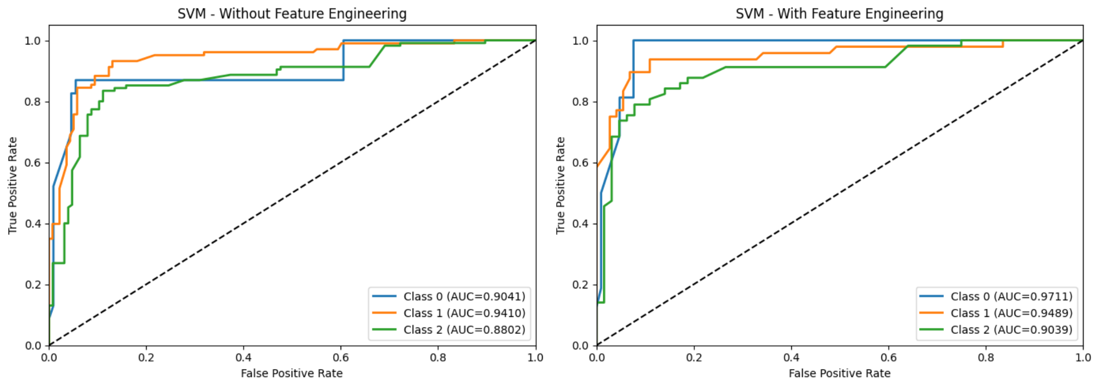
As part of the comprehensive evaluation, the performance of all models was assessed using five key classification metrics: precision, recall, F1-score, accuracy, and ROC AUC. The evaluation compared model performance before and after SHAP-based feature engineering, as summarized in Table 4.

Table 4. Comparison of model performance without and with feature engineering

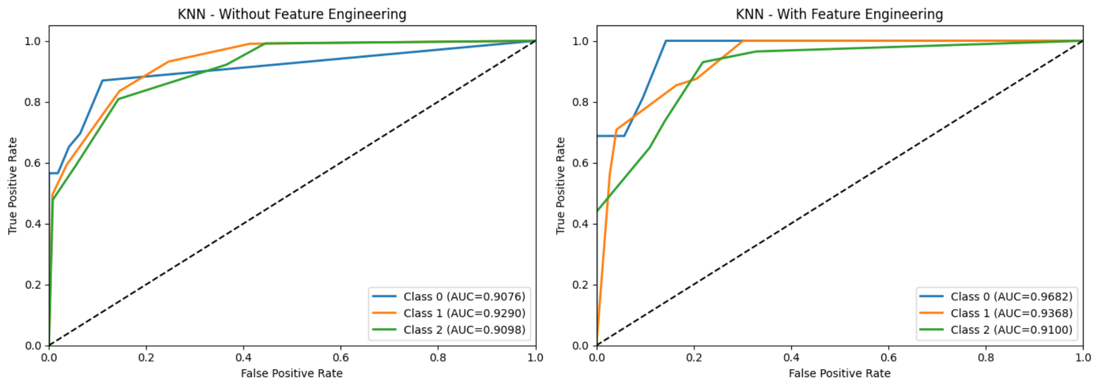
Model	Precision (Macro)	Recall (Macro)	F1-Score (Macro)	Accuracy	ROC AUC
SVM	74.04%	82.49%	75.36%	80.50%	90.87%
KNN	75.81%	76.53%	76.14%	80.50%	91.55%
Random Forest	83.73%	89.07%	85.80%	89.63%	98.04%
Decision Tree	82.78%	88.14%	84.87%	88.38%	95.08%
SVM + Feature Engineering	79.70%	87.28%	81.08%	82.64%	93.58%
KNN + Feature Engineering	74.81%	75.95%	75.21%	77.69%	93.83%
Random Forest + Feature Engineering	87.52%	92.76%	89.34%	90.08%	98.49%
Decision Tree + Feature Engineering	87.52%	92.76%	89.34%	90.08%	98.78%

Table 4 demonstrates that SHAP-based feature engineering led to performance improvements in most models, particularly Random Forest and Decision Tree, which both achieved the highest accuracy of 90.08%, along with an F1-score of 89.34% and ROC AUC values approaching 99%. The SVM model also showed gains, especially in recall, which rose from 82.49% to 87.28%. The only performance drop occurred in KNN, with a slight decline in accuracy and F1-score, although its ROC AUC improved. These results indicate that feature engineering enhances data representation and boosts predictive performance, particularly in tree-based models.

ROC curves were used to evaluate how well each model differentiated between classes before and after feature engineering. A higher AUC value reflects stronger classification performance. The visualization is presented in Figure 7.

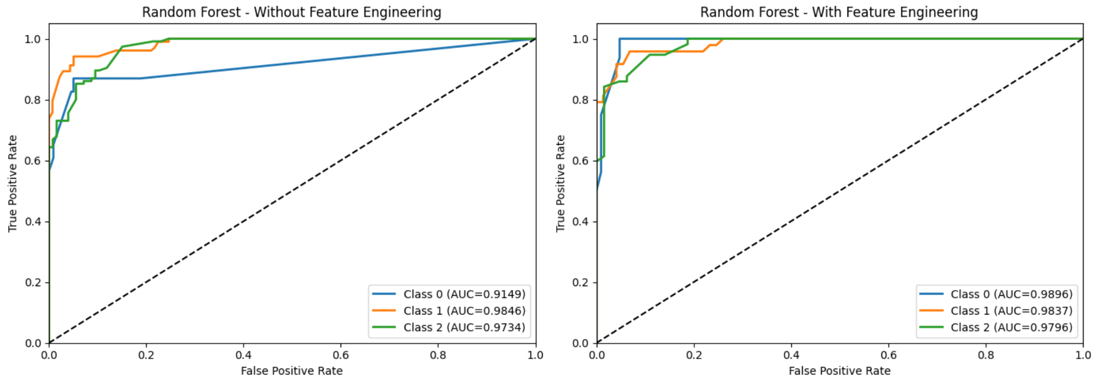


(a)

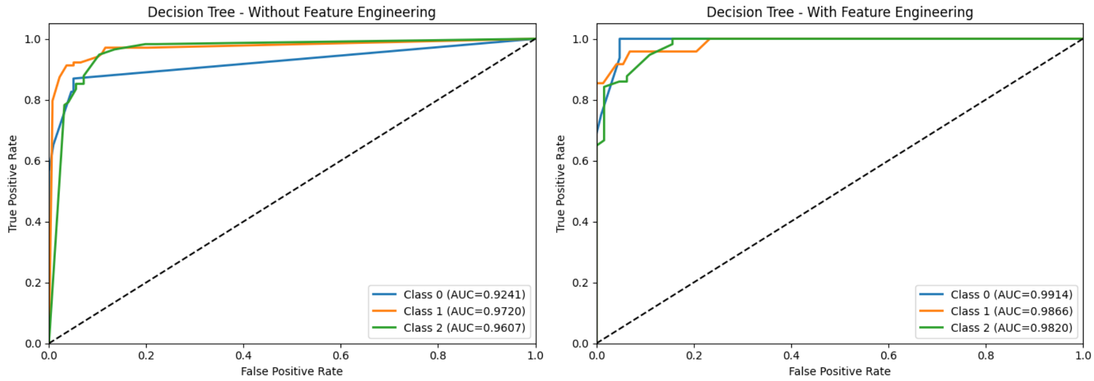


(b)

.



(c)



(d)

Figure 7. ROC curves of the models before and after applying feature engineering

The ROC curve results show that all models experienced an increase in AUC across all classes after feature engineering was applied. The most notable improvement was observed in the SVM model, where the average AUC increased from approximately 0.91 to 0.94. Meanwhile, Random Forest and Decision Tree, which already had high AUC values, further improved and approached 0.99. These results indicate that the addition of SHAP-based engineered features not only improved overall accuracy but also enhanced the models' ability to distinguish each target class with greater precision.

In addition to predictive performance, the computational trade-off was evaluated by measuring the average training time for each model in both the baseline and SHAP-based feature engineering scenarios. The training time was measured in the Google Colab environment using the *time.perf_counter()* function in Python, which recorded the duration taken by the *fit()* method for each model. To ensure stability and reduce the effect of random fluctuations, each model was trained three times, and the mean training time was reported. The results of the average training time before and after feature engineering are shown in Table 5.

Table 5. Average training time before and after feature engineering

Model	Avg. Training Time (Baseline)	Avg. Training Time (FE)	Difference
SVM	0.3316 s	0.3299 s	-0.0017 s
KNN	0.0028 s	0.0009 s	-0.0019 s
Random Forest	0.2221 s	0.2342 s	+0.0121 s
Decision Tree	0.0035 s	0.0042 s	+0.0007 s

The results in Table 5 show that the inclusion of SHAP-based engineered features had a negligible impact on training time for all models. Tree-based models such as Random Forest and Decision Tree experienced minor increases of approximately 0.0121 seconds and 0.0007 seconds, respectively. Surprisingly, non-tree models like SVM and KNN exhibited slightly faster training times in the feature engineering scenario. This could be attributed to the newly created features providing more informative representations, enabling SVM to converge faster and slightly reducing the computational overhead in KNN's data storage process. Overall, the performance improvements achieved through SHAP-based feature engineering were not accompanied by any significant increase in computational cost, indicating that the proposed approach remains efficient for practical deployment.

To verify whether the observed performance differences between the baseline and feature engineering scenarios were statistically significant, a paired t-test was conducted on the accuracy scores obtained from 10-

fold stratified cross-validation for each model. This approach ensures that the evaluation considers variability across different train-test splits and that the comparison is made on matched folds. The resulting t-statistics, p-values, and significance outcomes are presented in Table 6.

Table 6. Paired t-test results for accuracy before and after feature engineering

Model	t-statistic	p-value	Significance
SVM	2.8741	0.0185	Significant
KNN	-1.5423	0.1570	Not Significant
Random Forest	1.2145	0.2523	Not Significant
Decision Tree	2.2769	0.0482	Significant

The results in Table 6 show that the performance improvement of the SVM and Decision Tree models is statistically significant at the 5% level, thus confirming that the accuracy improvement is unlikely to be due to random variation. In contrast, the changes observed for KNN and Random Forest are not statistically significant, suggesting that the differences in their accuracy before and after feature engineering could be attributed to sampling variability. These findings highlight that the proposed SHAP-based feature engineering approach provides clear and measurable benefits for certain algorithms, particularly margin-based and hierarchical classifiers.

3.5. Discussion and Implications

The evaluation results indicate that applying SHAP-based feature engineering has a positive impact on the performance of most classification models, particularly tree-based algorithms such as Random Forest and Decision Tree. This suggests that the engineered features derived from SHAP value analysis are able to capture meaningful interactions between variables that are often missed in raw input formats. For instance, the engineered feature *Edu_Duration*, which combines Education Level and Class Duration, reflects a nuanced relationship between the time invested in online learning and the educational stage of students in determining adaptability.

Random Forest and Decision Tree models demonstrated the most significant improvements in F1-score and ROC AUC, confirming the ability of ensemble and hierarchical classifiers to leverage added information from SHAP-guided transformations. Interestingly, the SVM model exhibited a notable increase in recall (from 82.49% to 87.28%) without a proportional improvement in F1-score or accuracy. This phenomenon suggests that the feature engineering process enabled the SVM to better identify more positive instances, particularly those previously misclassified, at the cost of precision. Since SVM is sensitive to the placement of the decision boundary, adding new features may have caused the model to generalize better to minority classes. This improved recall but slightly reduced the balance between precision and recall, leading to a modest gain in F1-score. A similar trend was observed in the KNN model, where recall marginally increased while accuracy and F1-score slightly declined. This could be attributed to KNN's reliance on local neighborhood structures, which can be disrupted when new features alter distance relationships between samples. The increase in ROC AUC, however, indicates that KNN benefited from better separation between classes in probability space, even though it struggled to make correct class assignments under hard decision boundaries. These findings highlight that different models respond differently to the same set of engineered features. While tree-based models can flexibly split based on new interactions, margin-based models such as SVM and instance-based models such as KNN might exhibit asymmetric improvements depending on how the new features affect class boundaries or neighborhood dynamics.

In addition to predictive performance, the practical aspect of computational cost was examined. Average training times measured over three runs showed negligible differences between the baseline and feature engineering scenarios, with variations of less than 0.02 seconds for all models. Interestingly, SVM and KNN recorded slightly faster training times after feature engineering, which could be due to the engineered features providing more informative representations that facilitated faster convergence for SVM and slightly simplified the data structure for KNN. These results suggest that the performance improvements did not come at the expense of substantial increases in computational requirements, making the proposed approach viable for real-world applications.

Furthermore, statistical significance testing using the paired t-test confirmed that the observed accuracy improvements for SVM and Decision Tree were significant at the 5% level ($p < 0.05$), while changes for KNN and Random Forest were not statistically significant. This finding reinforces that the benefits of SHAP-based feature engineering are more pronounced for certain model types, particularly margin-based and hierarchical classifiers, and that observed gains for other models may be within the range of random variation.

A major strength of this study lies in its integration of explainable AI techniques (via SHAP) with semantically meaningful and statistically justified feature engineering. This approach not only improves model performance but also enhances transparency and trust in the algorithmic decision-making process. Nonetheless, this study has some limitations. First, the number of engineered features is still limited and may not capture all complex nonlinear interactions. Second, the evaluation was conducted on a single dataset from a specific educational context, which may restrict generalizability. Future research should explore a wider range of models, including neural networks, and validate findings on diverse educational datasets to confirm the robustness and scalability of the proposed approach. Additionally, future studies should implement more extensive cross-validation schemes and test the models on datasets with different sample sizes, class distributions, and educational contexts to better assess the external validity of the proposed method.

4. CONCLUSION

This study proposed the use of SHAP (SHapley Additive exPlanations) not only as a tool for model interpretability but also as a foundation for feature engineering in the classification of students' adaptability to online learning. By combining SHAP values with domain-driven reasoning, three new features were constructed to capture critical aspects of the learning context: the interplay between education level and class duration, the quality of digital infrastructure through device and internet type, and the students' socioeconomic conditions. Experimental results demonstrated that SHAP-based feature engineering consistently enhanced model performance, particularly for tree-based algorithms. Both Random Forest and Decision Tree models achieved the highest accuracy of 90.08%, F1-score of 89.34%, and ROC AUC values approaching 99%. The paired t-test confirmed that the accuracy improvements for SVM and Decision Tree were statistically significant at the 5% level ($p < 0.05$), while changes for KNN and Random Forest were not significant, suggesting that the benefits of SHAP-based feature engineering are more pronounced for certain model types. In addition, the training time analysis indicated that the computational cost remained negligible across all models, with variations below 0.02 seconds, ensuring that the performance gains were achieved without compromising efficiency. These findings highlight the practical potential of SHAP-guided feature engineering in improving the quality and fairness of student adaptability prediction models. Educational institutions could apply this approach to develop adaptive learning support systems that identify students who are struggling or highly adaptable, enabling timely and targeted interventions. By increasing model interpretability and precision, such systems could foster more inclusive and data-driven decision-making in online education settings. However, this study is not without limitations. The feature engineering process involved only a small number of manually constructed features, and the evaluation was limited to a single dataset from a specific educational context with relatively homogeneous demographic and learning environment characteristics. These factors may constrain the generalizability of the results to broader or different educational settings. For future research, it is recommended to explore automated feature generation techniques, conduct more extensive cross-validation, and test the approach across larger and more diverse educational datasets from different contexts. Further exploration of interpretability techniques in conjunction with fairness metrics could also support the development of responsible AI in education.

CONFLICT OF INTEREST STATEMENT

The Authors state no conflict of interest.

REFERENCES

[1] R. Wu and Z. Yu, “Relationship between university students’ personalities and e-learning engagement mediated by achievement emotions and adaptability,” *Educ. Inf. Technol.*, vol. 29, no. 9, pp. 10821–10850, 2024, doi: 10.1007/s10639-023-12222-5.

[2] S. P. Kar, A. K. Das, R. Chatterjee, and J. K. Mandal, “Assessment of learning parameters for students’ adaptability in online education using machine learning and explainable AI,” *Educ. Inf. Technol.*, vol. 29, no. 6, pp. 7553–7568, 2024, doi: 10.1007/s10639-023-12111-x.

[3] S. A. Salloum, A. Salloum, R. Alfaisal, A. Basiouni, and K. Shaalan, “Predicting Student Adaptability to Online Education Using Machine Learning,” in *Breaking Barriers with Generative Intelligence (BBGI)*, 2024, pp. 187–196. doi: 10.1007/978-3-031-65996-6_16.

[4] R. Arifudin, S. Subhan, and Y. N. Ifriza, “Student Adaptability Level Optimization using GridsearchCV with Gaussian Naive Bayes and K-Nearest Neighbor Methods as an Effort to Improve Online Education Predictions,” *J. Nas. Pendidik. Tek. Inform. JANAPATI*, vol. 14, no. 2, pp. 287–295, 2025, doi: 10.23887/janapati.v14i2.88972.

[5] M. M. H. Suzan, N. A. Samrin, A. A. Biswas, and M. A. Pramanik, “Students’ Adaptability Level Prediction in Online Education using Machine Learning Approaches,” in *International Conference on Computing Communication and Networking Technologies (ICCCNT)*, IEEE, 2021, pp. 1–7. doi: 10.1109/ICCCNT51525.2021.9579741.

[6] O. Iparraguirre-Villanueva *et al.*, “Comparison of Predictive Machine Learning Models to Predict the Level of Adaptability of Students in Online Education,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 4, pp. 494–503, 2023, doi: 10.14569/IJACSA.2023.0140455.

[7] R. Diallo, C. Edalo, and O. O. Awe, “Machine Learning Evaluation of Imbalanced Health Data: A Comparative Analysis of Balanced Accuracy, MCC, and F1 Score,” in *Practical Statistical Learning and Data Science Methods*, O. O. Awe and E. A. Vance, Eds., Cham: Springer Nature Switzerland, 2025, pp. 283–312. doi: 10.1007/978-3-031-72215-8_12.

[8] S. Wang, Y. Dai, J. Shen, and J. Xuan, “Research on expansion and classification of imbalanced data based on SMOTE algorithm,” *Sci. Rep.*, vol. 11, no. 1, pp. 1–11, 2021, doi: 10.1038/s41598-021-03430-5.

[9] S. Yang and S. Zhu, “Identifying factors influencing online learning outcomes for middle-school students — a re-examination based on XGBoost and SHAP,” *Educ. Inf. Technol.*, vol. 30, no. 11, pp. 15071–15094, 2025, doi: 10.1007/s10639-025-13405-

y.
[10] N. Bosch, “AutoML Feature Engineering for Student Modeling Yields High Accuracy, but Limited Interpretability,” *J. Educ. Data Min.*, vol. 13, no. 2, pp. 55–79, 2021, doi: 10.5281/zenodo.5275314.
[11] H. Wang, Q. Liang, J. T. Hancock, and T. M. Khoshgoftaar, “Feature selection strategies: a comparative analysis of SHAP-value and importance-based methods,” *J. Big Data*, vol. 11, no. 1, 2024, doi: 10.1186/s40537-024-00905-w.
[12] J. C. J. Luza and C. Rodriguez, “Predictive Attributes in Machine Learning for University Academic Performance: A Feature Engineering Approach,” in *IEEE International Conference on Computational Intelligence and Communication Networks*, IEEE, 2024, pp. 443–456. doi: 10.1109/CICN63059.2024.10847424.
[13] D. Hooshyar and Y. Yang, “Problems With SHAP and LIME in Interpretable AI for Education: A Comparative Study of Post-Hoc Explanations and Neural-Symbolic Rule Extraction,” *IEEE Access*, vol. 12, no. October, pp. 137472–137490, 2024, doi: 10.1109/ACCESS.2024.3463948.
[14] J. Faouzi and O. Colliot, “Classic Machine Learning Methods,” in *Machine Learning for Brain Disorders*, O. Colliot, Ed., New York, NY: Springer US, 2023, pp. 25–75. doi: 10.1007/978-1-0716-3195-9_2.
[15] M. M. H. Suzan and N. A. Samrin, “Students Adaptability Level in Online Education,” Kaggle. [Online]. Available: https://www.kaggle.com/datasets/mdmahmudulhasansuzan/students-adaptability-level-in-online-education
[16] I. O. Muraina, “Ideal Dataset Splitting Ratios in Machine Learning Algorithms: General Concerns for Data Scientists and Data Analysts,” in *International Mardin Artuklu Scientific Researches Conference*, 2022, pp. 496–505.
[17] T. Wongvorachan, S. He, and O. Bulut, “A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining,” *Information*, vol. 14, no. 1, 2023, doi: 10.3390/info14010054.
[18] N. A. Azhar, M. S. M. Pozi, A. M. Din, and A. Jatowt, “An Investigation of SMOTE Based Methods for Imbalanced Datasets With Data Complexity Analysis,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 6651–6672, 2023, doi: 10.1109/TKDE.2022.3179381.
[19] Z. L. Chia, M. Ptaszynski, F. Masui, G. Leliwa, and M. Wroczynski, “Machine Learning and feature engineering-based study into sarcasm and irony classification with application to cyberbullying detection,” *Inf. Process. Manag.*, vol. 58, no. 102600, 2021, doi: https://doi.org/10.1016/j.ipm.2021.102600.
[20] Y. Guan, F. Wang, and S. Song, “Interpretable machine learning for academic performance prediction: A SHAP-based analysis of key influencing factors,” *Innov. Educ. Teach. Int.*, pp. 1–20, doi: 10.1080/14703297.2025.2532050.
[21] L. C. Nnadi, Y. Watanobe, M. M. Rahman, and A. M. John-Otumu, “Prediction of Students’ Adaptability Using Explainable AI in Educational Machine Learning Models,” *Appl. Sci.*, vol. 14, no. 12, p. 5141, 2024, doi: 10.3390/app14125141.
[22] B. Etaati, A. Jahangiri, G. Fernandez, M. Tsou, and S. G. Machiani, “Understanding Active Transportation to School Behavior in Socioeconomically Disadvantaged Communities: A Machine Learning and SHAP Analysis Approach,” *Sustainability*, vol. 16, no. 1, p. 48, 2023, doi: 10.3390/su16010048.
[23] W. Chow, “Improving early dropout detection in undergraduate students: Exploring key predictors through SHAP values,” *Proceedings of the 35th Annual Conference of the Australasian Association for Engineering Education (AAEE 2024)*. Engineers Australia, Christchurch, New Zealand, Dec. 18, 2024. [Online]. Available: https://search.informit.org/doi/10.3316/informit.T2025032000014092035132356
[24] H. Yang, W. Lee, and J. Kim, “Identification of Key Factors Influencing Teachers’ Self-Perceived AI Literacy: An XGBoost and SHAP-Based Approach,” *Appl. Sci.*, vol. 15, no. 8, 2025, doi: 10.3390/app15084433.
[25] N. Syam and R. Kaul, “Random Forest, Bagging, and Boosting of Decision Trees,” in *Machine Learning and Artificial Intelligence in Marketing and Sales*, Emerald Publishing Limited, 2021, pp. 139–182. doi: 10.1108/978-1-80043-880-420211006.
[26] H. Liu, X. Chen, and X. Liu, “Factors influencing secondary school students’ reading literacy: An analysis based on XGBoost and SHAP methods,” *Front. Psychol.*, vol. 13, no. September, 2022, doi: 10.3389/fpsyg.2022.948612.
[27] S. S. Shanto and A. I. Jony, “Interpretable Ensemble Learning Approach for Predicting Student Adaptability in Online Education Environments,” *Knowledge*, vol. 5, no. 2, p. 10, 2025, doi: 10.3390/knowledge5020010.
[28] A. Aldino, A. Saputra, A. Nurkholis, and S. Setiawansyah, “Application of Support Vector Machine (SVM) Algorithm in Classification of Low-Cape Communities in Lampung Timur,” *Build. Informatics, Technol. Sci.*, vol. 3, no. 3 SE-Articles, Dec. 2021, doi: 10.47065/bits.v3i3.1041.
[29] D. Kurniadi, A. Mulyani, and I. Muliana, “Prediction System for Problem Students using k-Nearest Neighbor and Strength and Difficulties Questionnaire,” *J. Online Inform.*, vol. 6, no. 1, pp. 53–62, 2021, doi: 10.15575/join.v6i1.701.
[30] A. Arista, “Comparison Decision Tree and Logistic Regression Machine Learning Classification Algorithms to determine Covid-19,” *Sinkron*, vol. 7, no. 1, pp. 59–65, 2022, doi: 10.33395/sinkron.v7i1.11243.
[31] M. Nachouki, E. A. Mohamed, R. Mehdi, and M. Abou Naaj, “Student course grade prediction using the random forest algorithm: Analysis of predictors’ importance,” *Trends Neurosci. Educ.*, vol. 33, no. 100214, 2023, doi: https://doi.org/10.1016/j.tine.2023.100214.
[32] M. S. Mohosheu, F. Abrar Shams, M. A. al Noman, S. R. Abir, and Al-Amin, “ROC Based Performance Evaluation of Machine Learning Classifiers for Multiclass Imbalanced Intrusion Detection Dataset,” in *International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, 2023, pp. 1–6. doi: 10.1109/ICRAIE59459.2023.10468177.
[33] A. M. Carrington *et al.*, “Deep ROC Analysis and AUC as Balanced Average Accuracy, for Improved Classifier Selection, Audit and Explanation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 329–341, 2023, doi: 10.1109/TPAMI.2022.3145392.

.

BIOGRAPHIES OF AUTHORS

Yasin Efendi is a lecturer at Universitas Muhammadiyah Jakarta (UMJ), Faculty of Educational Sciences, in the Information Technology Education Study Progra. He is also actively involved as a board member of the Ikatan Asesor Professional (IASPRO) DKI Jakarta and the Lembaga Sertifikasi Profesi (LSP). He earned his master's degree from Sekolah Tinggi Teknologi Informasi (STTI) Benarif Jakarta, majoring in Informatics Engineering, and completed his doctoral studies at Universitas Negeri Jakarta (UNJ), majoring in Educational Technology. He has held several academic positions, including Head of the Program at AMIK Serang, Vice Director I for Academic Affairs at STMIK Serang, Head of the Informatics Management Study Program and Vice Director I for Academic Affairs at AMIK Wahana Mandiri, Pondok Cabe, South Tangerang, and Head of the Informatics Engineering Study Program at STTI NIIT I-TECH. His other activities include community service through training and coaching for youth communities, educators, and other social organizations. These efforts reflect his dedication to the academic community and his principle of living a life that brings benefit to others around him. He can be contacted at email: yasin.efendi@umj.ac.id.