Available online at website: https://jurnal.iaii.or.id/index.php/RESTI

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 10 No. 1 (2026) 1 - 10 e-ISSN: 2580-0760

Leveraging Machine Learning to Predict Academic Specialization Pathways in Higher Education

Rendy Wirawan Tamrin[1], Wella[2]*
[1,2]Department of Information System, Faculty of Engineering and Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia

[1]rendy.wirawan@student.umn.ac.id, [2]wella@umn.ac.id

Abstract

This study developed a machine learning-based model to predict academic concentration selection among information systems students at Universitas Multimedia Nusantara (UMN). A survey of 125 students from the 2024 cohort revealed that 90% experienced difficulties in choosing a specialization, primarily due to limited information on course relevance, unclear academic pathways, and career uncertainty. While the survey provides a contextual background, the predictive model was trained using historical academic performance data from the 2021–2023 cohorts. The three classification algorithms, Decision Tree, Random Forest, and Extreme Gradient Boosting (XGBoost) were implemented following the CRISP-ML methodology. To address class imbalance in the dataset, the Synthetic Minority Over-sampling Technique (SMOTE) was applied, followed by hyperparameter tuning and feature selection. The Random Forest model demonstrated superior performance, achieving an accuracy of 78.08% on the 2021–2022 cohort data, outperforming Decision Tree and XGBoost across all experimental settings. This result highlights Random Forest's robustness in this context, particularly after the integration of SMOTE and optimization procedures. The main contribution of this study lies in the application of machine learning for academic pathway prediction in an Indonesian higher education setting, providing a data-driven decision support tool to assist students in making informed and personalized specialization choices.

Keywords: decision tree; random forest; specialization prediction; XGBoost

How to Cite: R. W. Tamrin and Wella, "Leveraging Machine Learning to Predict Academic Specialization Pathways in Higher Education", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 10, no. 1, pp. 1 - 10, Feb. 2026.
Permalink/DOI: https://doi.org/10.29207/resti.v10i1.6941

Received: July 10, 2025
Accepted: October 18, 2025
Available Online: February 14, 2026

This is an open-access article under the CC BY 4.0 License
Published by Ikatan Ahli Informatika Indonesia

1. Introduction

The advancement of technology has significantly transformed various aspects of education, particularly through the integration of digital systems that enhance teaching, learning, and academic administration. This transformation is influenced by various personal and contextual factors that affect how technology is adopted across different educational environments [1]. Higher education institutions have traditionally relied on manual systems for managing academic information. Nowadays, digital integration is essential to improve efficiency, security, and data-driven decision-making [2]. Key academic resources such as curricula and syllabi are often only available through direct consultations or physical documents. This poses challenges, especially in selecting specialization courses that suit students' interests and competencies. Therefore, institutions must provide accessible and detailed information to support informed decision-making in specialization choices [3]. As institutions embrace digital transformation, they seek to improve educational services, including academic decision support, to remain relevant and competitive in a rapidly changing digital landscape [2].

At Universitas Multimedia Nusantara (UMN), the Information Systems undergraduate program offers three specialization tracks, which are Big Data Analytics, Database, and Enterprise Resource Planning [4]. Students in this program are required to choose a concentration track during their study [5]. These specializations are structured to match industry demands and provide domain-specific competencies. However, the current dissemination of information is limited to the program's official website or direct inquiries with the academic office, which has been found insufficient to support informed decision-making

[6]. This lack of a data-driven academic recommendation system at UMN creates a gap in student support services, particularly for specialization selection based on students' actual academic performance.

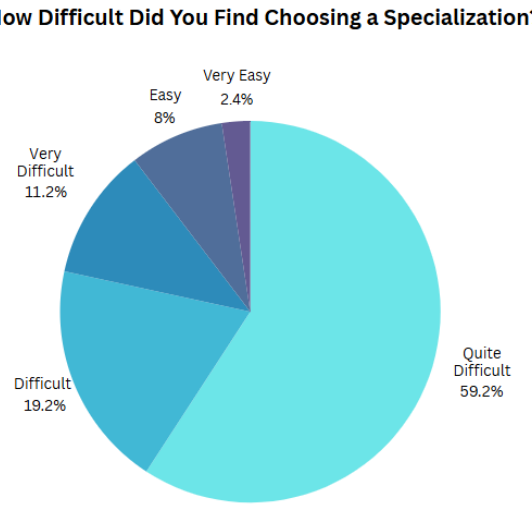


Figure 1. Difficulty Level in Choosing a Specialization

Figure 1 shows a survey result on the perceived difficulty in selecting a specialization among Information Systems students at UMN. A survey involving 125 Information Systems students from the 2024 cohort at UMN revealed that 89.6% found it difficult to choose a specialization. The primary reasons included a lack of understanding about the content and relevance of each track, uncertainty regarding future career prospects, and insufficient guidance connecting academic performance to suitable specialization options.

Given these challenges, it is necessary to implement a system that assists students in making data-informed decisions based on their academic performance. A recommendation system that leverages machine learning classification algorithms can serve as a solution to guide students objectively in selecting the most suitable specialization track. Previous studies have highlighted the potential of data-driven approaches to assist students in choosing academic concentrations. For example, a study analyzed students' concentration interests in Informatics using data-driven decision-making, emphasizing the importance of academic data for more objective recommendations [7]. Similarly, mapping student profiles for predicting their academic preferences based on past cohorts has been explored to help institutions better understand student tendencies [8].

Among various machine learning approaches, decision tree-based algorithms such as Decision Tree, Random Forest, and XGBoost have been widely used in educational data mining due to their accuracy and interpretability [9]. Decision Trees offer a clear hierarchical structure that makes it easy to trace the logic behind recommendations, while Random Forest and XGBoost improve generalization through ensemble learning and gradient boosting, respectively [10], [11]. However, the performance of these algorithms is highly context dependent. Research has shown that accuracy may vary depending on the dataset, feature selection, preprocessing techniques, and hyperparameter tuning [10], [12]. In the context of specialization recommendation, few studies have systematically compared these models while integrating data preprocessing techniques such as class balancing and hyperparameter tuning.

This study aims to develop a machine learning-based recommendation system that predicts suitable specialization tracks for Information Systems students at Universitas Multimedia Nusantara based on their academic performance. Specifically, it evaluates and compares the performance of three algorithms such as Decision Tree, Random Forest, and XGBoost, by evaluating the accuracy and cross-validation scores. The development process includes data preprocessing, class balancing, and hyperparameter tuning to ensure model robustness. The expected output is a functional system that helps students make better-informed decisions regarding their specialization, thereby reducing uncertainty and improving alignment with their academic strengths and future goals.

Numerous studies have explored the application of machine learning algorithms in developing predictive and recommendation systems, particularly within the field of education. Kadam et al. [11] introduced a university recommendation system using Decision Tree, Random Forest, and XGBoost, where XGBoost demonstrated the highest accuracy of 81%, outperforming the others at 76%. Zayed et al. [12] compared Decision Tree, SVM, and Random Forest to recommend undergraduate programs, showing that Random Forest achieved superior performance (97.70%) after employing feature importance analysis and hyperparameter tuning. In the e-commerce domain, Purnamasari [13] emphasized the effectiveness of XGBoost in product recommendation systems, where it surpassed Gradient Boosting, Random Forest, K-Nearest Neighbor, and Decision Tree with an accuracy of 89.44%. Hakkal and Lahcen [14] further validated XGBoost's robustness, revealing its advantage over Logistic Regression in predicting student performance, particularly when utilizing k-fold cross-validation and hyperparameter optimization.

This study addresses this gap by developing a machine learning-based recommendation system for specialization selection in the UMN Information Systems program, focusing on academic performance as the primary input variable. Specifically, it evaluates and compares the performance of Decision Tree, Random Forest, and XGBoost algorithms using real student data from the 2021–2023 cohorts. The methodological contribution lies in the integration of Synthetic Minority Over-sampling Technique (SMOTE), hyperparameter tuning, and feature

elimination to enhance model robustness in an imbalanced academic dataset.

Unlike prior research that mainly focuses on general program recommendations or career guidance, this study provides a specialization-level recommendation system customized for UMN's curriculum structure. The expected outcome is a decision support tool that helps students align their academic track selection with their competencies, reducing uncertainty and improving academic and career alignment.

2. Methods

This research adopts the CRISP-ML framework, a framework designed to guide machine learning practitioners in systematically developing high-quality machine learning applications. CRISP-ML stands for Cross-Industry Standard Process for Machine Learning. It is a process model that adapts the well-known CRISP-DM methodology, which was originally designed for data mining to the specific requirements and challenges of machine learning development. CRISP-ML provides a structured lifecycle that guides practitioners from initial business understanding through to long-term operational use of ML systems [15]. This study adopts CRISP-ML because it was developed to address common challenges in industrial machine learning projects, such as lack of process guidance and data quality issues by providing a robust, tool- and domain-neutral framework that supports the development, deployment, and monitoring of ML models [16]. The framework is inspired by the levels of driving automation and is operationalized through the CRISP-ML process [17].

By integrating quality assurance checkpoints and promoting transparency across all project stages, CRISP-ML(Q) ensures that machine learning models are not only accurate during development but also reliable and maintainable in production environments. This makes it highly applicable for both academic and industrial contexts, where the ability to deliver sustainable, interpretable, and operational ML solutions is increasingly critical [17].

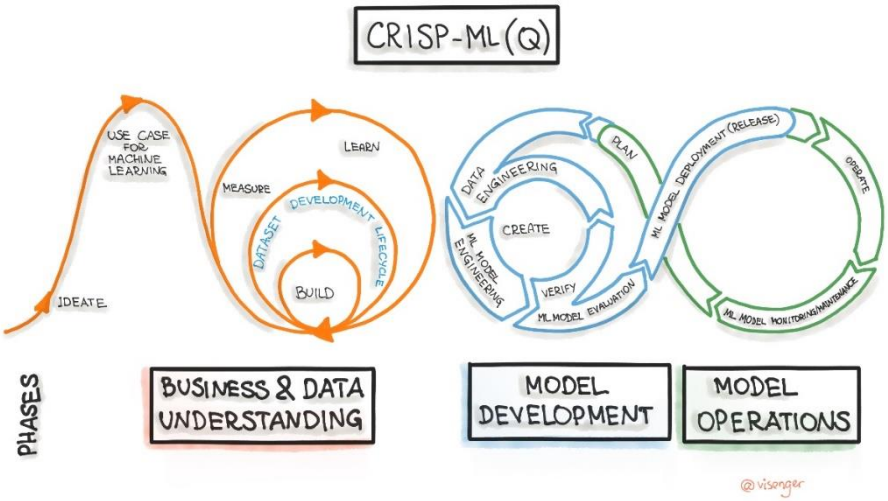


Figure 2. The Lifecycle Stages of the CRISP-ML Framework [15].

As shown in Figure 2, the CRISP-ML framework structured in six stages consists of business & data understanding, data engineering (data preparation), machine learning model engineering, evaluation, deployment, and monitoring & maintenance.

2.1 Business and Data Understanding

Limited access to and understanding of concentration course information among students in the Information Systems Program at Universitas Multimedia Nusantara has led to difficulties in selecting a concentration aligned with their interests and academic capabilities. A survey revealed that 90% of students from the 2024 cohort faced challenges in choosing concentration, primarily due to a lack of understanding about the course content and career prospects. This issue is further exacerbated by minimal academic guidance and limited integration of technology in decision-making processes.

To address this, the study aims to develop a machine learning-based concentration prediction system that provides data-driven recommendations to assist students in making more informed academic decisions based on their academic performance. The main objective is to build a predictive model for student concentration choices in the Information Systems Program at UMN using academic performance data. This model is intended to support students struggling to identify suitable concentrations and provide insights for academic advisors to enhance student guidance.

The dataset used comprises academic records of Information Systems students from cohorts 2021 to 2023, including GPA, grades for completed courses, and chosen concentration as the target variable. Data analyses are conducted to understand the data distribution, identify anomalies, handling missing values, and combining datasets.

2.2 Data Engineering (Data Preparation)

Following business and data understanding, the next phase is data preparation. This stage involves tasks such as data cleaning, integrating data from multiple sources if necessary, transforming data, and selecting relevant features. The goal is to ensure the dataset is ready for model training.

During this phase, several key preprocessing and modeling steps were conducted in a systematic manner. First, the dataset was examined to identify missing values, which were addressed either by removing incomplete records or by applying appropriate imputation techniques, such as mean substitution. Next, relevant features were selected based on their contribution to model performance, using feature importance analysis to retain variables that had a significant influence on the predictive outcome. To mitigate the issue of class imbalance in the target variable, the Synthetic Minority Over-sampling Technique (SMOTE) was applied, ensuring a more balanced distribution of samples across classes. Subsequently, hyperparameter tuning was carried out to determine the optimal parameter configurations for the selected algorithms, with the objective of maximizing model accuracy. Finally, the dataset was partitioned into training and testing subsets using a 70:30 ratio to enable reliable and unbiased model evaluation.

2.3 Machine Learning Model Engineering

This phase involves the design of the recommendation system model and the application of machine learning techniques. Three classification algorithms that are employed in this stage are Decision Tree, Random Forest, and XGBoost, which are commonly used for classification tasks due to their ability to handle complex, non-linear relationships in data.

Decision Tree. Decision Tree is a machine learning algorithm commonly used for classification and regression tasks [18]. In this study, decision trees are utilized due to their structured and interpretable approach in modeling decision processes. The concept originated from Clopper and Pearson in 1934, who introduced binary decision processes [19]. Significant advancements in decision tree methods were made by Breiman through the CART algorithm in 1984, which introduced binary splitting and the Gini index, and further developed by Quinlan with the introduction of ID3 in 1986 and its enhancement, C4.5, in 1993 [20]. These algorithms laid the foundation for decision trees to become a core element in ensemble methods such as random forests and boosting.

The training of a decision tree involves recursively splitting data into subsets based on criteria like information gain, Gini impurity, or variance reduction [21]. This process continues until a stopping condition such as maximum depth or node purity is met. According to Piramuthu, model performance heavily relies on data quality, making preprocessing crucial to remove missing values and outliers [20]. Furthermore, effective feature selection and engineering are essential to improve the accuracy and efficiency of the model. In shorts, it works by recursively splitting the dataset based on feature values, forming a tree structure where internal nodes represent attributes, branches represent decision rules, and leaves represent final predictions.

One widely used variant, C4.5, selects the best attribute for each split using Entropy and Information Gain. Entropy measures impurity or uncertainty in the data and is defined as Equation 1.

$$Entropy\ (S) = \sum_{i=1}^{c} p_i \log 2^p i \quad (1)$$

S refers to the dataset or sample under consideration. The term n represents the total number of distinct classes in that dataset. Each pi denotes the proportion or probability of data points belonging to class i within S. This probability is computed as the number of instances in class i divided by the total number of instances in the dataset.

Information Gain quantifies the reduction in entropy after a dataset is split on a particular attribute, as shown in Equation 2.

$$Gain\ (S, A) = \sum_{v \in v\,(A)} \frac{|S_v|}{|S|}\ Entropy(S_v) \quad (2)$$

In the Information Gain formula, S denotes the entire dataset, and A is the attribute being assessed. The term Entropy(S) represents the entropy of the whole dataset before splitting. Each v in Values(A) corresponds to a possible value of attribute A, and Sv is the subset of S where A takes that value. The notation $|Sv|$ indicates the number of samples in Sv, while $|S|$ is the total number of samples in the dataset. The ratio $|Sv|/|S|$ gives the proportion of samples in Sv, and Entropy(Sv)measures the entropy of that subset after the split.

Decision Tree models are easy to interpret and handle both numerical and categorical data. However, they are prone to overfitting and instability, especially when the tree is too deep or the dataset is imbalanced.

Random Forest is an ensemble-based machine learning algorithm that combines multiple decision trees to enhance prediction accuracy [22] [23]. It constructs a large number of decision trees using different random subsets of the training data and aggregates their outputs, typically through majority voting for classification or averaging for regression. Each decision tree in a random forest is trained on a bootstrap sample, meaning it is built using data randomly drawn with replacement from the original dataset [24]. Furthermore, Random Forest can be executed in parallel, offering computational advantages when dealing with large-scale data [25].

At each node, Random Forest commonly uses the Gini Index to measure impurity. The Gini Index quantifies the probability of incorrectly classifying a randomly chosen element and is defined as Equation 3.

$$\sum \sum j \neq i \left(\frac{f(CiT)}{|T|}\right)\left(\frac{f(CiT)}{|T|}\right) \quad (3)$$

$f(CiT)$ is the number of instances in class C_i within dataset T. Random Forest offers several advantages,

including robustness to overfitting, the ability to handle missing values, and support for both numerical and categorical data. However, it generally requires longer computation time than a single decision tree and produces models that are less interpretable due to the ensemble structure.

XGBoost (Extreme Gradient Boosting) is a powerful boosting algorithm that builds models sequentially, where each new model attempts to correct the errors made by its predecessors [22]. XGBoost, introduced by Chen and Guestrin, represents a major advancement in gradient boosting by combining flexibility, scalability, and computational efficiency through a novel regularization method that reduces overfitting and improves generalization [26]. It also enhances predictive accuracy by optimizing the gain function used for selecting split points within decision trees [26]. The algorithm optimizes a defined objective function composed of two key components: a loss function that quantifies prediction errors and a regularization term that controls model complexity to avoid overfitting.

The prediction, for instance i at iteration t is computed as Equation 4.

$$y_i^{(t)} = \sum_{k=1}^{t} \int k\ (\chi i) = y_i^{(t-1)} + \int t\ (\chi i) \quad (4)$$

$\int k$ represents the decision tree added at the k-th iteration, and χi is the feature vector of instance i. XGBoost offers key advantages such as computational efficiency, scalability to large datasets, and effective overfitting control via regularization. However, it requires more complex hyperparameter tuning and is generally harder to interpret than models like Random Forest or Decision Tree.

2.4 Evaluation and Deployment

After model development, the next phase is evaluation, which assesses the performance of the predictive models using evaluation metrics such as accuracy, recall, precision, and F1-score. This evaluation not only compares the performance of the three algorithms but also examines the impact of performance enhancement techniques such as feature selection, SMOTE, and hyperparameter tuning.

After the model has been evaluated and shows a low error rate, the next step is deployment. In this study, the deployment phase involves integrating the recommendation system into a web-based application to make it accessible and user-friendly for students. The application is developed using Python, a widely used programming language known for its simplicity, cross-platform compatibility, and extensive support for data science and machine learning tasks [27], [28]. Python provides high-level data structures, dynamic typing, and a large ecosystem of libraries such as Pandas, NumPy, and Scikit-learn, which are essential in building end-to-end ML pipelines [29]. Its interpreted nature allows developers to execute code directly without compilation, streamlining the development process [28].

To build the web application, this study uses Streamlit, an open-source Python framework designed for rapidly creating interactive web interfaces specifically for data science and machine learning projects [30]. Streamlit offers an intuitive and minimal-code approach to transform machine learning models into real-time, shareable applications. It can be installed via pip and integrates well with development tools like Visual Studio Code, enabling seamless development and debugging [30]. By utilizing Streamlit, the recommendation system can be deployed as a web application that allows students to interact directly with the model and receive program recommendations based on their input in real-time.

2.5 Monitoring and Maintenance

The predictive model requires regular monitoring. In this study, this stage involves analyzing the model's limitations and identifying opportunities for future development. Suggestions for further research include testing the model on future student cohorts or integrating non-academic data to enhance recommendation accuracy.

3. Results and Discussions

This section presents the experimental setup and results of the proposed student specialization prediction model. The experiments were conducted to evaluate the performance of three classification algorithms, consist of Decision Tree, Random Forest, and XGBoost using academic performance data of Information Systems students at Universitas Multimedia Nusantara. Several preprocessing and enhancement techniques, including SMOTE for handling class imbalance, hyperparameter tuning, and feature selection, were applied to optimize the model performance. The evaluation focuses on accuracy, precision, recall, and F1-score across different experimental scenarios.

3.1 Baseline Modeling

The initial modeling process was conducted using student data from the 2021–2023 cohorts, which was split into training and testing sets to train and evaluate the models. The classification algorithms applied in this stage were Decision Tree, Random Forest, and XGBoost. The evaluation results from this phase are presented in Table 1.

Table 1. Baseline Model Evaluation

Class	Precision	Recall	F1-Score	Model
0	0.75	0.65	0.70	Decision Tree
1	0.11	0.11	0.11	Decision Tree
2	0.35	0.53	0.42	Decision Tree
Accuracy		0.5429		Decision Tree
0	0.71	1.00	0.83	Random Forest
1	0.50	0.06	0.10	Random Forest
2	1.00	0.07	0.12	Random Forest
Accuracy		0.7048		Random Forest
0	0.69	0.89	0.78	XGBoost
1	0.29	0.11	0.16	XGBoost
2	0.20	0.07	0.10	XGBoost
Accuracy		0.6381		XGBoost

Among the three algorithms, the Random Forest algorithm achieved the highest accuracy at 0.7048. However, based on the precision, recall, and F1-score metrics, it is evident that the models tended to predict class 0 correctly more frequently, while struggling to accurately predict classes 1 and 2. This issue is likely due to the imbalanced nature of the dataset.

3.2 SMOTE (Synthetic Minority Over-sampling Technique)

Initial modeling was conducted using baseline models on a dataset comprising 370 student records. However, evaluation results revealed that the precision, recall, and F1-score were suboptimal and inconsistent for classes 1 and 2, indicating the presence of class imbalance. This imbalance poses a major challenge in predicting students' specialization preferences, as one or two classes tend to dominate significantly over the others.

To address this issue, the SMOTE technique was applied to balance the class distribution. The results after applying SMOTE are presented in Table 2.

Table 2. Performance Evaluation After Applying SMOTE

Class	Precision	Recall	F1-Score	Model
0	0.68	0.50	0.58	Decision Tree
1	0.21	0.33	0.26	Decision Tree
2	0.17	0.27	0.21	Decision Tree
Accuracy		0.44		Decision Tree
0	0.68	0.71	0.69	Random Forest
1	0.21	0.17	0.19	Random Forest
2	0.25	0.27	0.26	Random Forest
Accuracy		0.55		Random Forest
0	0.70	0.64	0.67	XGBoost
1	0.21	0.22	0.22	XGBoost
2	0.30	0.40	0.34	XGBoost
Accuracy		0.53		XGBoost

As shown in Table 2, the application of SMOTE led to improved performance metrics for the minority classes (classes 1 and 2), particularly in the XGBoost model. However, overall model accuracy decreased. This decline is likely due to the introduction of synthetic data, which can generate noise and make it difficult for the model to distinguish between real and synthetic instances. As a result, the model may experience slight overfitting in minority classes and reduced generalization performance.

3.3 SMOTE + Hyperparameter Tuning

Table 3 presents the results after applying hyperparameter tuning. The findings indicate that tuning did not lead to a significant improvement in model accuracy. The marginal gains suggest that the models' learning capacity is constrained by the quality and representation of the input features. This highlights that training data quality plays a more critical role in model performance than parameter adjustments alone.

Table 3. Model Accuracy with Hyperparameter Tuning

Model	Accuracy Before Tuning	Accuracy After Tuning
Decision Tree	0.44	0.4571
Random Forest	0.55	0.6000
XGBoost	0.53	0.5429

Consequently, while hyperparameter tuning is a well-established method for model optimization, its effectiveness is highly dependent on the underlying data representation. In this case, the limited performance improvement may be due to insufficiently informative features or a lack of clear predictive patterns within the data. Additionally, high class overlap may have hindered the models' ability to distinguish between classes, indicating the need for more relevant features or a larger training dataset to achieve more accurate predictions.

3.4 SMOTE + Hyperparameter Tuning + Feature Selection

To further improve model performance, feature selection was performed based on feature importance scores, along with data segmentation by academic cohort, specifically, the 2021-2022 and 2023 cohorts. These strategies were applied in conjunction with SMOTE and hyperparameter tuning to maintain consistency across experimental setups. The goal of feature selection was to eliminate less relevant or redundant attributes, thus improving the quality of input features and enhancing model generalization. Meanwhile, cohort-based segmentation aimed to account for potential variations in academic behavior and specialization preferences between different student batches, allowing the models to learn more targeted patterns within each cohort.

The feature selection technique based on feature importance was carried out by selecting the top five features with the highest importance scores to retrain the models.

Table 4. Model Accuracy with Feature Selection

Model	Acccuracy Before	Accuracy After
Decision Tree	0.4571	0.4476
Random Forest	0.6000	0.5619
XGBoost	0.5429	0.5619

The results, as shown in Table 4, indicate that only XGBoost experienced an improvement in accuracy, while Random Forest and Decision Tree showed a decline in performance. The decrease in accuracy for Random Forest and Decision Tree suggests that these algorithms heavily rely on tree complexity and depth, which require a greater number of informative features to generate accurate decisions. In contrast, XGBoost demonstrated the ability to adapt to a reduced feature set, provided the selected features are of high quality. This is attributed to its boosting approach, which allows it to learn from limited yet relevant information more effectively.

These findings highlight that feature selection based on feature importance should be applied selectively and with consideration of the specific characteristics of the algorithm being used. Inappropriate application may lead to degraded model performance, particularly for models that depend on rich feature representation.

After multiple attempts to improve model accuracy using various enhancement techniques, the results still did not reach optimal performance. Therefore, the next experiment was conducted by segmenting the dataset based on academic cohorts, specifically, students from the 2021-2022 and 2023 cohorts. This segmentation was conducted because of the curriculum differences across cohorts, which were suspected to contribute to the model's suboptimal accuracy. The objective of this experiment was to isolate the influence of curriculum variation and data heterogeneity and to evaluate whether greater homogeneity in curriculum structure and data consistency could lead to improved model accuracy.

As shown in Table 5, the results indicate that the Random Forest algorithm achieved the highest accuracy of 0.7808, followed by XGBoost at 0.6986, and Decision Tree at 0.6027. This improvement is likely driven by several factors. The first factor is the data is more stable with the curriculum structure across the 2021-2022 cohort. The class distribution looks relatively balanced, which is good to reduce bias during training. And the data volume is larger and more representative sample size, allowing the model to capture predictive patterns more effectively. The dataset from this cohort exhibits consistent coverage of core courses and sufficient data volume, which facilitates better pattern recognition by the models. These characteristics help improve the correlation between features and targets, ultimately leading to more accurate predictions.

Table 5. Performance Evaluation Based on the 2021-2022 Students Cohort Data

Model	Acccuracy
Decision Tree	0.6027
Random Forest	0.7808
XGBoost	0.6986

In contrast, the experiment using only the 2023 cohort data (see Table 6) showed a noticeable decline in model accuracy across all algorithms. Several factors may account for this performance drop, because of a smaller dataset size, limiting the model's generalization capability. It then lowers the academic variability, as most students were still in early semesters. And ongoing curriculum transitions, leading to inconsistencies in data structure compared to previously learned patterns.

These findings underscore that data homogeneity and curriculum stability are critical success factors in developing reliable academic specialization prediction models. Segmenting by cohort, particularly those with consistent academic structures, can significantly enhance model performance and reliability.

Table 6. Performance Evaluation Based on the 2023 Students Cohort Data

Model	Acccuracy
Decision Tree	0.4531
Random Forest	0.6250
XGBoost	0.5625

3.5 Discussion

This study conducted a series of experiments by implementing several tree-based machine learning algorithms, combined with various performance-enhancing techniques. The primary objective was to find the best model performance by evaluating the effectiveness of each approach in predicting students' academic specialization preferences based on their academic performance data. The experiments were performed through multiple configurations, starting from baseline models to the application of SMOTE, hyperparameter tuning, feature selection, and dataset segmentation based on academic cohorts.

Table 7 presents a summary of the experimental results using different techniques and dataset configurations.

Table 7. Summary of The Experimental Results Using Different Techniques

Dataset	Technique Applied	Decision Tree	Random Forest	XGBoost
2021–2023 Cohorts	Baseline	0.4705	0.7048	0.6248
2021–2023 Cohorts	SMOTE	0.4571	0.6000	0.5429
2021–2023 Cohorts	SMOTE + Hyperparameter Tuning	0.4571	0.6000	0.5429
2021–2023 Cohorts	SMOTE + Tuning + Feature Selection	0.4476	0.5619	0.5619
2021–2022 Cohort (split)	SMOTE + Hyperparameter Tuning	0.6027	0.7808	0.6986
2023 Cohort (split)	SMOTE + Hyperparameter Tuning	0.4531	0.6250	0.5625

Table 7 summarizes the performance of the three tree-based models across six experimental configurations. In the first experiment, the baseline model using data from 2021 to 2023 showed that Random Forest achieved the highest accuracy at 0.7048, followed by XGBoost and Decision Tree. However, the models struggled to classify minority classes accurately, indicating a class imbalance issue. The application of SMOTE slightly improved class distribution but did not enhance accuracy, likely due to noise introduced by synthetic data. Hyperparameter tuning provided stability but limited gains, suggesting that data quality played a more significant role. Feature selection improved XGBoost's performance but led to reduced accuracy in the other models.

To further explore model performance, the dataset was segmented by student cohort. Training the models on the 2021-2022 cohort significantly improved accuracy, with Random Forest reaching 0.7808, attributed to curriculum consistency, balanced class distribution, and a more representative dataset. In contrast, the 2023 cohort yielded lower accuracy due to smaller sample size, limited academic variation, and ongoing

curriculum transitions. These results indicate that model success is not solely dependent on algorithm selection, but also on thoughtful data preprocessing, segmentation, and alignment with educational structures. Despite some limitations, the experiments offer valuable insights for designing an academic recommendation system using machine learning.

To understand the position of this research, we compared it with several previous studies that successfully applied machine learning algorithms to similar recommendation and classification systems. Table 8 is a comparison between this research and several previous studies.

Table 8. Comparison of Model Accuracy with Previous Research

Ref	Algorithm	Accuracy	Research Purpose
[11]	XGBoost	81%	College Predictions
[12]	Random Forest	97.70%	Undergraduate Program Recommendations
[13]	XGBoost	89.44%	E-Commerce Product Recommendations
[14]	XGBoost	79.9%	Student Performance Predictions
This Research	Random Forest	78.08%	Predicted Interests of UMN Information Systems Students

Based on Table 8, The findings of this study indicate that the Random Forest algorithm achieved the highest performance in predicting student interest in the Information Systems Study Program at Multimedia Nusantara University (UMN), reaching an accuracy of 78.08% for the 2021–2022 cohort after the application of SMOTE and hyperparameter tuning. Although this accuracy is lower than that reported by Zayed et al. [12], who employed Random Forest for undergraduate program recommendations and achieved 97.70%, the discrepancy may be attributed to factors such as the complexity and diversity of predictor variables, dataset size, and the distinct characteristics of the educational domains involved.

When compared with studies utilizing XGBoost, such as those by Kadam et al. [11], Purnamasari [13], and Hakkal and Lahcen [14], the accuracy achieved in the present study remains relatively competitive, particularly given the institutional-scale dataset and its inherent class imbalance. The integration of SMOTE and parameter tuning in this research substantially enhanced the model's performance relative to the baseline, even though it did not surpass the highest accuracies reported in the literature. These results underscore that the optimal choice of algorithm is inherently dependent on the specific characteristics of the dataset and the intended predictive objectives.

To present a student specialization prediction solution for the Information Systems program at UMN, this study deploys the best model which has the highest accuracy through a user-friendly and real-time web-based application. As said before, this application utilizes only the best-performing model (Random Forest), which was selected based on evaluation results showing the highest accuracy among all tested models.

Figure 3 illustrates the Streamlit-based application for predicting student specializations in the Information Systems program at UMN.

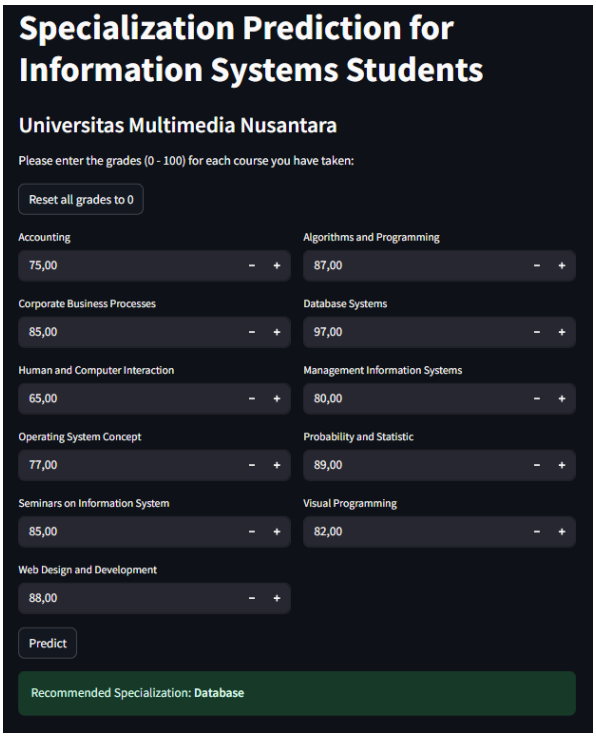


Figure 3. User Interface of the Student Specialization Prediction System

The application accepts input in the form of course grades, processes the data in accordance with the trained feature structure, and outputs a predicted specialization label which consists of Big Data Analytics, Database, or Enterprises Resource Planning. The application allows users, such as students, lecturers, or academic staff to input grades from mandatory courses already taken by the student. These grades are processed according to the feature order and structure used during model training. It is important to note that each input grade directly influences the prediction result, as machine learning models like Random Forest rely heavily on the data patterns learned from the training set. As a result, inaccurate or low grades may affect the prediction quality.

4. Conclusions

This study developed a machine learning-based predictive model for academic concentration selection among Information Systems students at Universitas Multimedia Nusantara, utilizing three tree-based classification algorithms: Decision Tree, Random Forest, and XGBoost. By incorporating Synthetic Minority Over-sampling Technique (SMOTE), hyperparameter tuning, and feature selection within the CRISP-ML framework, the Random Forest model achieved the highest accuracy of 78.08% on the 2021–

2022 cohort after data segmentation. Key academic features such as Mobile Application Development and Data Visualization emerged as significant predictors, demonstrating the potential of academic performance data to guide specialization recommendations. While the accuracy achieved is promising, it also indicates room for improvement to ensure more reliable and precise recommendations for students. Model performance in this context is influenced not only by algorithm selection but also by data representativeness, curriculum consistency, and preprocessing strategies.

Future research should consider integrating additional features beyond academic grades, such as behavioral data, learning styles, or psychological profiles, to capture a more holistic view of student capabilities and preferences. Moreover, the application of more advanced ensemble methods or deep learning approaches could further enhance model accuracy and robustness. Expanding the dataset longitudinally and across multiple institutions would also help validate the generalizability of the proposed system, supporting broader implementation in academic decision support.

Acknowledgements

The author expresses sincere gratitude to the Department of Information Systems at Universitas Multimedia Nusantara for their support throughout the research process. Special thanks are also extended to the academic advisors who guided, provided valuable feedback and encouragement.

References

[1] A. Cattaneo, M. L. Schmitz, P. Gonon, C. Antonietti, T. Consoli, and D. Petko, "The role of personal and contextual factors when investigating technology integration in general and vocational education," *Comput Human Behav*, vol. 163, no. May 2024, 2025, doi: 10.1016/j.chb.2024.108475.

[2] A. Marks and M. Al-Ali, "Digital Transformation in Higher Education: A Framework for Maturity Assessment," *COVID-19 Challenges to University Information Technology Governance*, vol. 11, no. 12, pp. 61–81, 2022, doi: 10.1007/978-3-031-13351-0_3.

[3] Made Pradnyana Ambara and I Nyoman Suraja Antarajaya, "Development of Academic Information Systems to Support the Lecture Process in Educational Institutions," *Jurnal Teknologi Informasi dan Komputer*, vol. 9, no. 3, pp. 291–297, 2023, doi: 10.36002/jutik.v9i3.2505.

[4] "Sistem Informasi | Universitas Multimedia Nusantara." Accessed: Apr. 29, 2025. [Online]. Available: https://www.umn.ac.id/sistem-informasi/

[5] Umn.ac.id, "To Become a Leading Global University in ICT & Entrepreneurship," 2024. [Online]. Available: https://www.umn.ac.id/profil/#:~:text=2005-,2005,yang

[6] D. A. N. Arah and K. Mahasiswa, "Socialization of Selection of Interest in The Islamic Economics Study Program as an Effort to Strengthen Students' Competencies and Career Direction," vol. 1, no. 1, pp. 55–64, 2025, doi: https://doi.org/10.71312/mrdimas.v1i1.243

[7] F. Mahardika, F. Supriadi, and A. Guntara, "Analysis of student interest distribution in informatics concentration using the data-driven decision-making approach," vol. 19, no. 2, pp. 278–287, 2025, doi: https://doi.org/10.33480/inti.v19i2.6347.

[8] H. Ar - Rasyid, S. F. Pane, and M. Y. H. Setyawan, "Student Profile Mapping to Predict Student Interests," *Petir*, vol. 16, no. 1, pp. 1–10, 2023, doi: 10.33322/petir.v16i1.1337.

[9] S. Gheewala, S. Xu, and S. Yeom, "Deep attention dynamic representation learning networks for recommender system review modeling," *Big Data Research*, vol. 40, no. March, p. 100521, 2025, doi: 10.1016/j.bdr.2025.100521.

[10] I. Cahyati *et al.*, "Implementation of Business Intelligence with Artificial Intelligence in E-Commerce," *SENTRI: Jurnal Riset Ilmiah*, vol. 3, no. 6, pp. 2741–2756, 2024, doi: 10.55681/sentri.v3i6.2904.

[11] Y. Kadam, Y. Kadulkar, V. Moolya, and S. Agrawal, "University Recommendation System for Abroad Studies using Machine Learning," *Int J Res Appl Sci Eng Technol*, vol. 11, no. 4, pp. 2954–2962, 2023, doi: 10.22214/ijraset.2023.50835.

[12] Y. Zayed, Y. Salman, and A. Hasasneh, "A Recommendation System for Selecting the Appropriate Undergraduate Program at Higher Education Institutions Using Graduate Student Data," *Applied Sciences (Switzerland)*, vol. 12, no. 24, 2022, doi: 10.3390/app122412525.

[13] A. M. Purnamasari, "E-Commerce Product Recommendations using XGBoost with User Clusters and Clickstream," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 4, pp. 5942–5948, 2020, doi: 10.30534/ijatcse/2020/258942020.

[14] S. Hakkal and A. A. Lahcen, "XGBoost To Enhance Learner Performance Prediction," *Computers and Education: Artificial Intelligence*, vol. 7, no. June, p. 100254, 2024, doi: 10.1016/j.caeai.2024.100254.

[15] "CRISP-ML(Q)." Accessed: May 10, 2025. [Online]. Available: https://ml-ops.org/content/crisp-ml

[16] S. Studer *et al.*, "Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology," *Mach Learn Knowl Extr*, vol. 3, no. 2, pp. 392–413, 2021, doi: 10.3390/make3020020.

[17] J. Rosemeyer, C. Neunzig, C. Akbal, J. Metternich, and B. Kuhlenkötter, "A maturity model for digital ML tools to be used in manufacturing environments", 2024, doi: https://doi.org/10.26083/tuprints-00026519.

[18] B. Charbuty and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, 2021, doi: 10.38094/jastt20165.

[19] C. Gupta and A. K. Ramdas, "Distribution-free calibration guarantees for histogram binning without sample splitting," pp. 1–25, 2021., doi: https://doi.org/10.48550/arXiv.2105.04656

[20] I. D. Mienye and N. Jere, "A Survey of Decision Trees : Concepts , Algorithms , and Applications," *IEEE Access*, vol. 12, no. June, pp. 86716–86727, 2024, doi: 10.1109/ACCESS.2024.3416838.

[21] Y. Sun and R. Quinlan, "Prediction performance of improved decision tree-based algorithms : Prediction performance of improved decision tree-based algorithms : a review," *Procedia Manuf*, vol. 35, pp. 698–703, 2019, doi: 10.1016/j.promfg.2019.06.011.

[22] H. Tan, "Machine Learning Algorithm for Classification," *J Phys Conf Ser*, vol. 1994, no. 1, 2021, doi: 10.1088/1742-6596/1994/1/012016.

[23] N. Younas, A. Ali, H. Hina, M. Hamraz, Z. Khan, and S. Aldahmani, "Optimal Causal Decision Trees Ensemble for Improved Prediction and Causal Inference," *IEEE Access*, vol. 10, pp. 13000–13011, 2022, doi: 10.1109/ACCESS.2022.3146406.

[24] I. D. Mienye, Y. Sun, and S. Member, "A Survey of Ensemble Learning : Concepts , Algorithms , Applications , and Prospects," *IEEE Access*, vol. 10, no. August, pp. 99129–99149, 2022, doi: 10.1109/ACCESS.2022.3207287.

[25] H. A. Salman, A. Kalakech, and A. Steiti, "Random Forest Algorithm Overview," vol. 2024, pp. 69–79, 2024.

[26] M. Imani, A. Beikmohammadi, and H. R. Arabnia, "Comprehensive Analysis of Random Forest and XGBoost Performance with SMOTE, ADASYN, and GNUS Under Varying Imbalance Levels," *Technologies (Basel)*, vol. 13, no. 3, pp. 1–40, 2025, doi: 10.3390/technologies13030088.

[27] "Apa itu Python? ? - Penjelasan Bahasa Python - AWS." Accessed: May 10, 2025. [Online]. Available: https://aws.amazon.com/id/what-is/python/

[28] "Library Python yang Populer di Kalangan Data Scientist - Algoritma." Accessed: May 10, 2025. [Online]. Available: https://algorit.ma/blog/library-python/
[29] O. N. Abdumalikovich, "Classification of Python Language Modules," *Indonesian Journal of Innovation Studies*, vol. 19, no. July, 2022, doi: http://dx.doi.org/10.21070/ijins.v19i.672.
[30] "Streamlit • A faster way to build and share data apps." Accessed: May 10, 2025. [Online]. Available: https://streamlit.io/