Enhancing Intelligent Tutoring Systems through SVM-Based Academic Performance Classification and Rule-Based Question Recommendation

*[1]Fenina Adline Twince Tobing, [2]Toto Haryanto
[*1,2]School of Data Science, Mathematics and Informatics, IPB University, Bogor, West Java, 16680, Indonesia
[1]Informatics Department, Faculty of Engineering and Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia
email: *[1]feninatobing@apps.ipb.ac.id, [1]fenina.tobing@umn.ac.id, [2]totoharyanto@apps.ipb.ac.id

Accepted 04 June 2025
Approved 09 June 2025

***Abstract*— This research aims to automatically classify students' academic performance levels using Support Vector Machine (SVM) algorithm and automatically recommend questions based on classification results. Dataset consists of six assignment scores per student, averaging students into three performance levels: Beginner, Intermediate, and Advanced. Before training, the data undergoes preprocessing involving normalization with StandardScaler and splitting into training and testing sets. Model is trained using Radial Basis Function (RBF) kernel with hyperparameter tuning to optimize its performance. Evaluation results show that the model achieved an accuracy of 91.67%, with a precision of 93.06%, a recall of 91.67%, and an F1-score of 91.89%. The best performance was found in Intermediate class, the dominant category in dataset, while performance in Advanced category was relatively lower due to limited sample size. Following classification, a rule-based recommendation system is used to suggest questions that match the student's predicted level of competence. This approach supports a more adaptive and personalized learning environment. The findings demonstrate, SVM algorithm effectively supports intelligent learning systems such as Intelligent Tutoring System (ITS). Future work should include data balancing techniques, expansion of dataset size, and comparative analysis with other algorithms such as Random Forest or K-Nearest Neighbors (KNN) to enhance model generalization.**

Index Terms*— Academic Performance; Intelligent Tutoring System (ITS); Machine Learning; Question Recommendation; Support Vector Machine (SVM).

I. INTRODUCTION

Technological advancements in education have significantly contributed to the emergence of Intelligent Tutoring Systems (ITS). AI driven platforms designed to deliver tailored instruction by analyzing students' learning behaviors and academic performance. ITS enables more adaptive and individualized learning processes, enhancing students' engagement and understanding [1]. One of the main challenges in ITS development is the system's capability to accurately assess and classify student academic performance. Traditional methods often rely on average test scores or assignment grades, which may not fully represent a student's learning trajectory or personal characteristics [2]. Such static assessments are less effective in dynamic learning environments that require timely and responsive instructional feedback.

Affective Intelligent Tutoring Systems (Affective ITS) have emerged as an advanced approach to personalize learning by incorporating students' emotional responses. As noted by Fernández-Herrero [3], such systems utilize emotion recognition to adapt feedback and content, thereby improving engagement and learning outcomes in real-time educational settings. Liu, Latif, and Zhai [4] conducted a systematic review highlighting recent developments in Intelligent and Robot Tutoring Systems. Their findings emphasize that AI-driven tutoring technologies enhance adaptability and student engagement, while also noting ongoing challenges in scalability, ethics, and cognitive modeling.

Machine Learning (ML) has emerged as a promising solution to this issue by offering objective and data-driven methods for performance analysis. Among various ML algorithms, Support Vector Machine (SVM) is particularly well-regarded for its accuracy and effectiveness in classification tasks. Prior studies [5], have shown that SVM models can accurately predict academic ability based on demographic data and learning styles. Likewise, research [6] demonstrated the application of hybrid AI models combining SVM and Decision Trees for real-time content recommendations in ITS, especially in STEM education. In addition to its widespread application in educational contexts, the Support Vector Machine (SVM) algorithm has also shown strong performance in text classification and sentiment analysis, [7] successfully implemented SVM combined with Chi-Square feature selection to categorize user feedback into sentiment classes, demonstrating the algorithm's effectiveness in handling large-scale data classification with a reported accuracy of 77%. This reinforces the suitability of SVM for high-dimensional and complex classification tasks, such as predicting student performance levels in adaptive learning environments like Intelligent Tutoring Systems (ITS).

Classify students' academic performance using assignment score data. Students are categorized into three performance levels—Beginner, Intermediate, and Advanced—based on the average of six assignments [8]. Dataset is preprocess through standardization, and model performance is evaluated using metrics such as accuracy, precision, recall, and F1-score [9] indicating strong performance in identifying student categories, particularly within the Intermediate group.

Following classification, the system employs a rule-based recommendation approach to suggest practice questions from a structured question bank. Questions are organized by difficulty level, allowing the system to match students with materials appropriate to their skill level. This integration of classification and recommendation supports personalized learning, enabling ITS to deliver more effective and targeted instruction. In summary, this research presents a hybrid framework combining SVM-based classification with rule-based question recommendation, supporting the broader goal of developing intelligent and adaptive educational systems. The proposed approach not only enhances the diagnostic capabilities of ITS but also improves relevance and impact of the learning content provided to students.

II. METHODOLOGY

This research employs SVM algorithm to perform academic performance classification. [10] explored the use of SVM to classify student learning abilities in system modeling and simulation courses. Their findings demonstrate that SVM-based classification supports effective personalization by aligning instructional content with learners' ability levels, thereby enhancing educational outcomes in technical learning environments.

The methodological workflow begins with data collection and preprocessing, which includes data cleaning, normalization, and splitting the dataset into training and testing subsets. SVM model is trained through a hyperparameter tuning process to optimize its performance. Model evaluation is conducted using a confusion matrix, along with performance metrics such as precision, recall, and F1-score, to assess the model's accuracy and classification effectiveness.

A. *Methodological Flowchart*

The overall process of the study is summarized in the methodological flowchart shown in figure 1, which outlines main steps from data preparation to personalized question recommendation based on classification outcomes

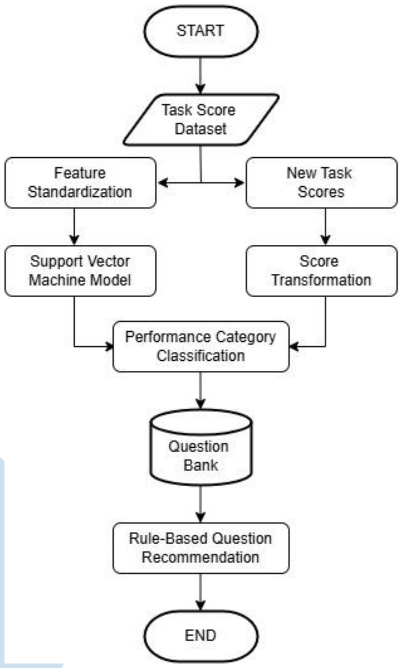

Fig1. Research Flowchart

To visualize the end-to-end process, a flowchart is presented in Figure 1, outlining each major stage in the research methodology—from data processing to the automated recommendation of questions based on classified student performance. The stages are as follows:

1. Start → The system begins with the initialization of the classification and recommendation modules.
2. Assignment Score Dataset → The system receives input in the form of student assignment scores, consisting of six recorded task values for each student.
3. Feature Standardization → These scores are standardized using the StandardScaler method to ensure consistent feature scaling, preventing any single feature from dominating the model training process.
4. SVM Model Training → The classification model is developed using SVM algorithm with a Radial Basis Function (RBF) kernel. The model is trained on the standardized data, with tuning applied to parameters C and gamma to enhance accuracy.
5. New Assignment Scores → The system accepts new input data representing assignment scores of unclassified students.

6. Score Transformation → These new scores undergo the same standardization process as the training data to ensure consistency during prediction.
7. Performance Classification → The standardized data is then classified by the trained SVM model into one of three academic performance levels: Beginner, Intermediate, or Advanced.
8. Question Bank → The system maintains a curated question bank categorized by difficulty levels and subject topics.
9. Rule-Based Question Recommendation → Based on the classification results, the system recommends questions aligned with the student's predicted performance level using a Rule-Based Recommendation strategy that maps student categories to question difficulty.
10. End → The process concludes once the student receives questions tailored to their level, supporting the implementation of a technology-driven adaptive learning system.

B. *RBF Kernel in SVM*

SVM is a machine learning algorithm that operates within a hypothesis space to identify optimal decision boundaries for classification tasks. SVM is widely used for both binary and multi-class classification problems, where it aims to separate data into distinct classes using a decision boundary known as a hyperplane. If the data is linearly separable, the hyperplane appears as a straight line; otherwise, for non-linear cases, the boundary is curved and more complex [11].

SVM relies on several key parameters that significantly influence the performance of the classification model, namely gamma, cost (C), and kernel function [12].

- Gamma parameter defines how far the influence of a single training example reaches. A low gamma value means that the influence extends far, producing smoother decision boundaries, while a high gamma indicates a more localized influence, which may capture more complex patterns but risks overfitting.
- Cost parameter (C) controls the trade-off between achieving a low error on training data and maintaining a simple decision boundary. A larger C places more emphasis on correctly classifying training examples, potentially at the cost of model generalization.
- Kernel function is responsible for transforming the input data into a higher-dimensional space where a linear separator may be found more easily. This process enables SVM to handle complex classification problems, even when the data is not linearly separable in its original space.

Several types of kernels are commonly used in SVM [13]:

1. Linear Kernel is the simplest form and is used when the dataset is linearly separable. It is computationally efficient and suitable for problems with high-dimensional but sparse features.
2. Polynomial Kernel is applied when the decision boundary is non-linear. It maps the original data into a higher-dimensional space using polynomial functions. This kernel includes a degree parameter (d), which controls the flexibility of the model. However, using a higher degree may lead to less stable performance.
3. RBF Kernel is particularly effective for non-linear datasets. It maps data into an infinite-dimensional space, allowing for complex decision boundaries. The gamma parameter in RBF plays a crucial role in defining the influence of each data point. Compared to other kernels, RBF tends to provide lower classification errors and better generalization in many applications.

RBF kernel is employed in this study due to its effectiveness in handling non-linearly separable data. In the context of academic performance classification, the relationship between features (such as assignment scores and other characteristics) and student performance categories is not always linear. The RBF kernel addresses this challenge by projecting input data into a higher-dimensional space where linear separation is more feasible. Moreover, the RBF kernel is known for its flexibility and empirical performance, making it a reliable choice for classification tasks. It includes the gamma parameter, which controls the influence of individual training samples and helps capture complex local patterns in the dataset. Given these advantages, the RBF kernel was selected to ensure optimal accuracy and efficiency in classifying student academic performance using the SVM model.

C. *Confusion Matrix*

Confusion Matrix is a common tool for evaluating performance machine learning models and applied binary as well as multiclass classification problems. As noted [14], it provides a tabular summary of prediction outcomes across four key categories as in Figure 2.

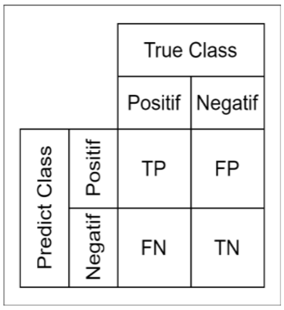

Fig 2. Confusion Matrix

Confusion Matrix provides a structured overview of a model's prediction performance by categorizing outcomes into four components: True Positive (TP) – correctly predicted positive instances; True Negative (TN) – correctly predicted negative instances; False Positive (FP) – incorrect positive predictions; and False Negative (FN) – incorrect negative predictions [15].

Based on these components, several evaluation metrics can be derived to assess classification performance:

1. Accuracy represents the overall proportion of correct predictions and is calculated using the formula:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

2. Precision measures the proportion of positive identifications that were actually correct, defined as:

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

3. Recall (also known as sensitivity) reflects the model's ability to retrieve all relevant instances and is given by:

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

4. F1-Score is the harmonic mean of precision and recall, useful for evaluating models with imbalanced classes:

$$F1score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (4)$$

These metrics provide a comprehensive understanding of how well a classification model performs, especially when dealing with imbalanced datasets.

III. Result And Discussion

Based on training results SVM model, classification performance on test dataset was found to be satisfactory. Model was evaluated using a confusion matrix along with performance metrics such as precision, recall, and F1-score. The results indicate that the model is capable of achieving high accuracy, particularly in the majority class. The obtained F1-score reflects a balanced trade-off between precision and recall, suggesting the model is not only accurate but also reliable in identifying the target categories.

Hyperparameter tuning process played a crucial role in enhancing the model's overall performance, especially in optimizing choice of kernel and regularization parameters. The best classification results were achieved using the Radial Basis Function (RBF) kernel with optimized values of C and gamma, underscoring the importance of careful parameter selection in improving the effectiveness of the SVM algorithm.

Table I delineates the distribution of subjects classified into distinct performance categories as determined by the applied classification methodology. This table offer structured summary of categorization outcomes utilized in analyses.

TABLE I. Category Distribution Table

Category	Count	Percentage
Intermediate	62	51.67%
Beginner	40	33.33%
Advanced	18	15.0%

Dataset utilized in this study is categorized into three levels of student performance: Beginner, Intermediate, and Advanced. The distribution of the data reveals that the Intermediate category is the most dominant, comprising 62 records (51.67%), followed by Beginner with 40 records (33.33%), and Advanced with only 18 records (15%). This class imbalance warrants attention, as it may affect the model's ability to accurately learn and identify instances from minority class—particularly the Advanced category, which contains the fewest data samples.

RBF kernel is widely used in SVM to address non-linear classification problems. Unlike linear kernels, RBF kernel maps data into a higher-dimensional space, allowing classes to be separated by a hyperplane. This transformation enables the capture of complex data patterns, which is critical in applications such as image recognition and medical diagnostics. RBF kernel computes the similarity between data points using a Gaussian function, where the γ (gamma) parameter controls the range of influence of each point. A higher γ results in narrower influence, while a lower γ leads to broader influence across the decision boundary. RBF kernel facilitates efficient computation without explicitly projecting data into the higher-dimensional space. This property reduces computational complexity and improves generalization to unseen data, thereby mitigating overfitting. Due to its flexibility and efficiency, the RBF kernel remains a preferred choice for SVM-based classification tasks involving complex and non-linear relationships.

RBF kernel is specifically for training the SVM model due to its effectiveness in dealing with non-linear data.

1. Feature Normalization:
   The begins with normalizing features using StandardScaler. This step is essential because SVMs are sensitive to the scale of the data. By applying StandardScaler, the data is transformed so that it has a mean of 0 and a standard deviation of 1. This ensures that all features contribute equally to the model training and that the RBF kernel performs optimally without being biased by differences in the scale of features.

2. Using RBF Kernel:
   a. SVC (Support Vector Classification) model is being created with the RBF kernel (svm_model = SVC(kernel='rbf')).
   b. RBF kernel is chosen because it excels at handling complex, non-linear relationships between data points. Unlike linear kernels, which can only separate data that is linearly separable, the RBF kernel maps the data into a higher-dimensional space, allowing SVM to find a hyperplane that can separate data points more effectively.

3. Why RBF Kernel:
   a. RBF kernel is especially suitable when dealing with complex patterns in the data that cannot be separated with a straight line or plane. It is also able to handle high-dimensional data, which is often the case in real-world scenarios. The kernel trick allows the SVM to compute the optimal hyperplane in this higher-dimensional space without explicitly transforming the data, making the process computationally efficient.
   b. In this context, using the RBF kernel allows model to create flexible decision boundaries, ensuring that it can classify the data points in a way that linear kernels would not be able to do effectively. This is why the RBF kernel is preferred in this example.

4. Training the SVM Model:
   Once data is scaled, SVM model is trained using RBF kernel. svm_model.fit(X_train_scaled, y_train) line indicates the model is learning from the training data (X_train_scaled) and their corresponding labels (y_train), with the RBF kernel ensuring that the decision boundaries can adapt to complex data patterns

The model becomes more flexible and can handle non-linear relationships between the data points, providing better performance on real-world datasets that often exhibit complex patterns. Table II presents the classification performance metrics used to evaluate the effectiveness of the proposed model across different learner categories.

TABLE II. TABLE CLASSIFICATION REPORT

Category	Precision	Recall	F1-Score	Support
Advanced	0.67	1.00	0.80	2
Beginner	1.00	0.83	0.91	6
Intermediate	0.94	0.94	0.94	16
accuracy	0.9306	0.9167	0.9189	24
macro avg	0.87	0.92	0.88	24
weighted avg	0.93	0.92	0.92	24

Trained SVM model was evaluated using a test set comprising 24 samples, including 16 Intermediate, 6 Beginner, and 2 Advanced instances. Evaluation results indicate that the model achieved an overall accuracy of 91.67%, with precision of 93.06%, recall of 91.67%, and an F1-score of 91.89%. Per-class classification performance reveals that the model performed best on the Intermediate category, achieving precision, recall, and F1-score values of 0.94. For the Beginner category, the model attained perfect precision (1.00) but a lower recall of 0.83, suggesting that one Beginner instance was misclassified. In the Advanced category, the model achieved a recall of 1.00, indicating it correctly identified all Advanced instances; however, the precision was only 0.67, suggesting that some samples from other categories were incorrectly label as Advanced. Overall, the model yielded a weighted average F1-score of 0.92, reflecting strong general classification performance despite some imbalance across class predictions. The high performance in the Intermediate class aligns with its dominance in the dataset.

Confusion matrix in Figure 3 further supports the evaluation findings by illustrating the distribution of predictions across the actual class labels

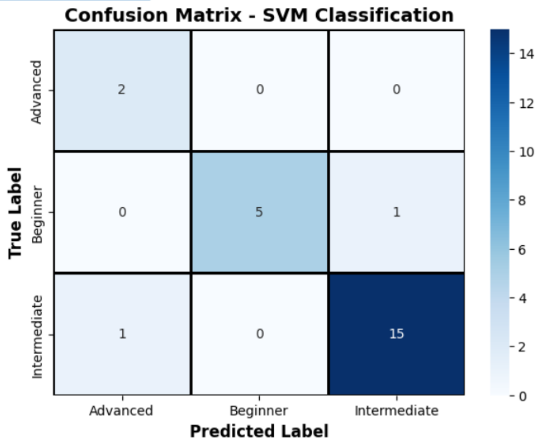

Fig 3. Confusion Matrix Analysis

All Advanced instances (2 samples) were correctly classified with no errors. In the Beginner class, 5 out of 6 samples were accurately identified, while 1 sample was misclassified as Intermediate, accounting for the imperfect recall value. In the Intermediate class, 15 out of 16 samples were correctly predicted, with 1 sample misclassified as Advanced. This indicates that the model occasionally confuses Intermediate with Advanced instances, although its overall performance in the dominant class remains high.

```python
import pandas as pd
import numpy as np

# Buat new_data sebagai DataFrame dengan nama kolom yang sama
new_data = pd.DataFrame(np.array([[10, 80, 75, 90, 88, 70]]), columns=X_train.columns)

# Transformasi dengan StandardScaler tanpa peringatan
new_data_scaled = scaler.transform(new_data)
kategori_prediksi = svm_model.predict(new_data_scaled)

print("\n✅ Kategori Prediksi untuk Data Baru:", kategori_prediksi[0])
```

```
✅ Kategori Prediksi untuk Data Baru: Intermediate
```

Fig 4. Student Performance Category Prediction using SVM

Figure 4 illustrates the prediction process of a new student's academic performance category based on their assignment scores. The input data is provided as a numerical array representing six assignment scores (e.g., 100, 80, 75, 90, 88, 70). These values are then converted into a Data Frame with the same structure and feature columns as the training dataset (X_train) to ensure compatibility with the model.

Prior to prediction, the input data is transformed using the pre-fitted StandardScaler to match the scale of the training data. This step ensures that feature distributions remain consistent between training and inference phases. The pre-trained SVM model (svm_model) then performs the classification on the standardized input.

The prediction result places the new student in the Intermediate category, indicating that based on the pattern of their assignment scores, the student is estimated to have a moderate academic performance level. This process demonstrates the practical application of classification models in automatically and objectively categorizing new students based on historical performance data.

Figure 5 showcases a selection of five randomly chosen questions from each academic performance level: Beginner, Intermediate, and Advanced. These questions are retrieved from a curated question bank that categorizes items based on difficulty and topic, such as HTML, CSS, JavaScript, and PHP. The sampling illustrates how the rule-based recommendation system works in practice, delivering learning materials aligned with the student's predicted skill level. By doing so, the system helps support more personalized and adaptive learning, ensuring that students engage with content that matches their current abilities.

```
Level yang tersedia: ['Beginner' 'Intermediate' 'Advanced']

🧠 Soal untuk Level 'Beginner':
    ID                                          Soal           Topik  \
8  Q18  Bagaimana cara membuat daftar tidak terurut (u...            HTML
1  Q02             Buatlah struktur dasar dokumen HTML.  HTML Structure
5  Q15  Apa itu tag <a> dan bagaimana cara membuat tau...            HTML
0  Q01  Apa itu HTML dan apa fungsinya dalam web devel...            HTML
7  Q17  Tulis kode HTML untuk menampilkan gambar denga...            HTML

      Tipe Soal     Level
8   Isian Singkat  Beginner
1  Praktik Coding  Beginner
5   Isian Singkat  Beginner
0   Isian Singkat  Beginner
7  Praktik Coding  Beginner

🧠 Soal untuk Level 'Intermediate':
     ID                                          Soal           Topik  \
18  Q23  Buat program JavaScript yang menampilkan alert...      JavaScript
11  Q07  Buatlah layout grid sederhana menggunakan Flex...      CSS Layout
15  Q20    Jelaskan konsep event handling pada JavaScript.      JavaScript
10  Q06     Apa itu responsive design dan mengapa penting?  CSS Responsive
17  Q22  Apa itu method 'querySelector()' dan bedanya d...  JavaScript DOM

         Tipe Soal         Level
18  Praktik Coding  Intermediate
11  Praktik Coding  Intermediate
15           Essay  Intermediate
10           Essay  Intermediate
17   Isian Singkat  Intermediate

🧠 Soal untuk Level 'Advanced':
     ID                                          Soal              Topik  \
28  Q29  Buat fungsi pencarian real-time dengan JavaScr...         Full Stack
21  Q12  Jelaskan bagaimana cara kerja AJAX dan manfaat...    JavaScript/AJAX
25  Q26  Jelaskan konsep asynchronous dalam JavaScript ...  JavaScript Async
20  Q11  Buat aplikasi sederhana (To-Do List) menggunak...         JavaScript
27  Q28  Implementasikan pagination pada daftar produk ...                PHP

         Tipe Soal     Level
28    Project Mini  Advanced
21           Essay  Advanced
25           Essay  Advanced
20    Project Mini  Advanced
27  Praktik Coding  Advanced
```

Fig 5. Question Bank Categorized by Difficulty Level

The question bank has been systematically categorized based on difficulty level to support the implementation of a recommendation system aligned with the students' classified performance. By leveraging the results from the SVM-based classification, the system is able to recommend questions that match the learner's current academic level Beginner, Intermediate, or Advanced thereby promoting a more adaptive and personalized learning experience

Each question entry includes several key attributes, such as:

- ID: A unique identifier assigned to each question.
- Question: The actual prompt or instruction that students must respond to or complete.
- Topic: The subject matter covered by the question, such as HTML, CSS, or HTML Structure.
- Question Type: The format in which the question is presented, including short answer, coding exercises, or multiple-choice.
- Level: The difficulty level of the question. In this sample, all items are label as Beginner, Intermediate, or Advanced.

Figure 6 implementation of automatic question recommendation system that operates following the classification of students' academic performance

```
🟦 Soal untuk Mahasiswa #1 (Kategori: Intermediate)

------------------------------
ID Soal     : Q06
Topik       : CSS Responsive
Tipe Soal   : Essay

------------------------------
ID Soal     : Q07
Topik       : CSS Layout
Tipe Soal   : Praktik Coding

------------------------------
ID Soal     : Q08
Topik       : HTML Form
Tipe Soal   : Essay

------------------------------
ID Soal     : Q09
Topik       : JS Form Validation
Tipe Soal   : Praktik Coding

------------------------------
ID Soal     : Q10
Topik       : JavaScript DOM
Tipe Soal   : Isian Singkat

------------------------------
ID Soal     : Q20
Topik       : JavaScript
Tipe Soal   : Essay
```

Fig 6. Automatically Recommended Questions Based on Student Classification

This recommendation process based on Rule-Based Recommendation approach, which aligns the performance level predicted by the SVM model with the corresponding difficulty level of questions from the pre-defined question bank.

For example, Student #1, who was classified as Intermediate by SVM model, receives a tailored set of questions that match their predicted performance level. Each recommended question includes key attributes such as the Question ID, Topic, and Question Type. The topics span various areas relevant to the Web Programming course, including CSS Responsive Design, CSS Layout, HTML Forms, JavaScript Form Validation, and JavaScript DOM Manipulation. The types of questions recommended include Essay, Coding Practice, and Short Answer, aimed at evaluating both conceptual understanding and practical application skills.

This rule-based matching system ensures that each student receives questions aligned with their performance category Beginner, Intermediate, or Advanced thereby fostering a more adaptive and personalized learning process. The goal of this approach is to enhance learning effectiveness while maintaining student engagement and motivation by delivering appropriately challenging content.

While using SVM and rule-based recommendation is not entirely new, this study brings a fresh perspective by focusing on real assignment scores as the primary features for classification, something not commonly used in similar works that often rely on broader data like demographics or exam scores. The system then connects the classification results directly to a structured question bank, offering students questions that truly match their learning level. What makes this approach different is its simple yet effective integration of machine learning with real-time instructional support, making it practical for classrooms or online learning environments. By combining these two components into one seamless process, students not only get categorized accurately but also receive personalized learning materials instantly. This tight coupling between prediction and recommendation adds real value to intelligent tutoring systems and helps move them closer to being truly adaptive and supportive in day to day learning.

IV. CONCLUSIONS

This research demonstrates SVM algorithm with RBF kernel can be effectively applied to classify students' academic performance into three categories: Beginner, Intermediate, and Advanced.

Preprocessing procedures most notably data normalization and dataset partitioning, contributed substantially to the improved performance of classification model. Evaluation results indicate the implemented Support Vector Machine (SVM) classifier achieved robust performance, with accuracy of 91.67%, precision of 93.06%, recall of 91.67%, and an F1-score of 91.89%. Among the classified categories, the model exhibited its highest performance in identifying the Intermediate group, which constituted the majority of instances in dataset. Conversely, lower performance was observed the Advanced category, potentially attributable to limited representation of samples in class.

Subsequent the classification process, SVM outputs were integrated into a Rule-based Question Recommendation mechanism. This component leveraged predicted performance categories to guide the selection of learning materials tailored to each learner's proficiency level. The findings collectively demonstrate the efficacy of combining machine learning-based classification with rule-based instructional support, underscoring the model's potential to enhance personalized learning within intelligent educational systems, particularly Intelligent Tutoring Systems (ITS).

For future work, several improvements are recommended:

- Data balancing techniques such as SMOTE (Synthetic Minority Over-sampling Technique) or undersampling can be applied to address class imbalance and improve performance on underrepresented categories.
- Alternative classification models such as Random Forest, K-Nearest Neighbors (KNN), and Gradient Boosting should be explored for comparative analysis to better understand their effectiveness on the same dataset.
- Expanding dataset size, especially for minority classes like Advanced, is crucial to

help the model learn better feature representations and improve generalization.

- Real-world application of the model in adaptive learning systems should also be considered, where the classified student level can guide the delivery of customized learning materials or assignments tailored to individual needs.
- Moreover, advanced evaluation strategies such as the Three-Way Confusion Matrix [14] may be incorporated to manage uncertain or borderline predictions. This would enhance the reliability of classification results and support more personalized decision-making within intelligent tutoring systems.

REFERENCES

[1] "AI in Education: The Rise of Intelligent Tutoring Systems | Park University." Accessed: Mar. 12, 2025. [Online]. Available: https://www.park.edu/blog/ai-in-education-the-rise-of-intelligent-tutoring-systems/

[2] C. C. Lin, A. Y. Q. Huang, and O. H. T. Lu, "Artificial intelligence in intelligent tutoring systems toward sustainable education: a systematic review," *Smart Learn. Environ.*, vol. 10, no. 1, Dec. 2023, doi: 10.1186/S40561-023-00260-Y.

[3] J. Fernández-Herrero, "Evaluating Recent Advances in Affective Intelligent Tutoring Systems: A Scoping Review of Educational Impacts and Future Prospects," 2024, doi: 10.3390/educsci14080839.

[4] V. Liu, E. Latif, and X. Zhai, "Advancing Education through Tutoring Systems: A Systematic Literature Review," Mar. 2025, Accessed: Apr. 22, 2025. [Online]. Available: https://arxiv.org/abs/2503.09748v1

[5] B. I. Nugroho, N. A. Santoso, and A. A. Murtopo, "Prediksi Kemampuan Akademik Mahasiswa dengan Metode Support Vector Machine," *remik*, vol. 7, no. 1, pp. 177–188, Jan. 2023, doi: 10.33395/REMIK.V7I1.12010.

[6] B. H. Rao, M. Rengarajan, R. Changala, K. Santosh, M. Aarif, and M. Ponni Valavan, "Hybrid AI Approach Combining Decision Trees and SVM for Intelligent Tutoring Systems in STEM Education," *10th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2024*, pp. 574–579, 2024, doi: 10.1109/ICACCS60874.2024.10716888.

[7] L. Luthfiana, J. C. Young, and A. Rusli, "Implementasi Algoritma Support Vector Machine dan Chi Square untuk Analisis Sentimen User Feedback Aplikasi," *Ultim. J. Tek. Inform.*, vol. 12, no. 2, pp. 125–126, 2020, doi: 10.31937/ti.v12i2.1828.

[8] S. Bulut, "A Comparison of the First Three Flute Lessons with Beginner and Intermediate-advanced Level Flute Students," *Procedia - Soc. Behav. Sci.*, vol. 177, pp. 229–234, Apr. 2015, doi: 10.1016/J.SBSPRO.2015.02.397.

[9] M. Armanda, F. Adline, T. Tobing, and U. M. Nusantara, "Implementation of Support Vector Machine Method for Twitter Sentiment Analysis Related to Cancellation of u-20 World Cup in Indonesia," vol. 11, no. 1, 2024.

[10] C. Liu, S. Yang, and M. Engineering, "Personalized Learning Ability Classification Using SVM for Enhanced Education in System Modeling and Simulation Courses," vol. 1, no. 4, pp. 295–307, 2024.

[11] N. Hendrastuty, A. Rahman Isnain, and A. Yanti Rahmadhani, "Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja Pada Twitter Dengan Metode Support Vector Machine," vol. 6, no. 3, 2021, Accessed: Mar. 12, 2025. [Online]. Available: http://situs.com

[12] A. Zaiem Praghakusma and N. Charibaldi, "Komparasi Fungsi Kernel Metode Support Vector Machine untuk Analisis Sentimen Instagram dan Twitter (Studi Kasus : Komisi Pemberantasan Korupsi)," vol. 9, no. 2, pp. 33–42, 2021, doi: 10.12928/jstie.v8i3.xxx.

[13] "Apa itu Kernel Trick? Pengertian dan Jenis-jenis Fungsi Kernel SVM - Trivusi." Accessed: Mar. 12, 2025. [Online]. Available: https://www.trivusi.web.id/2022/04/fungsi-kernel-svm.html

[14] J. Xu, Y. Zhang, and D. Miao, "Three-way confusion matrix for classification: A measure driven view," *Inf. Sci. (Ny).*, vol. 507, pp. 772–794, Jan. 2020, doi: 10.1016/J.INS.2019.06.064.

[15] S. Sathyanarayanan and B. R. Tantri, "Confusion Matrix-Based Performance Evaluation Metrics," no. November, 2024, doi: 10.53555/AJBR.v27i4S.4345