**Joinforcs: Journal of informatics and information system**
**Volume 1, (4) 2026, p. 139-149**
**E-ISSN : XXXX-XXXX**
**©2026. The Author(s). This is an open acces article under cc-by-sa**

# Comparative Analysis of K-Nearest Neighbor (KNN) and Decision Tree Algorithms in Predicting Student Academic Performance

Muhammad Iqbal Tsabitul Azmi[a,1], Dimas Andreas Musthofa [a,2], Aldo Paristiyo[a,3], Muh. Rizky Perdana[a,4] , Muhammad Hanif K.[a,5], Tommy Dwi Putra[a,6]
[a]Department of Informatics, STMIK AMIKOM Surakarta, Indonesia
[1]muhammad.130521@mhs.amikomsolo.ac.id, [2]dimas.130518@mhs.amikomsolo.ac.id, [3]aldo.130543@mhs.amikomsolo.ac.id, [4]muh.130537@mhs.amikomsolo.ac.id, [5]muhammad.130538@mhs.amikomsolo.ac.id, [6]tommy@dosen.amikomsolo.ac.id*
* corresponding author

**Article Info**

***Article History:***

Received: July 28, 2026
Revised: August 4, 2026
Accepted: August 12, 2026

***Keywords :***
*Class Imbalance, Decision Tree, Educational Data Mining, K-Nearest Neighbor, Performance Prediction*

*This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License*

***Abstract***

*Predicting students' academic performance is essential for identifying learning needs and providing timely educational interventions. Although various machine learning algorithms have been applied in Educational Data Mining (EDM), comparative studies evaluating their performance on datasets that integrate both academic and non-academic factors remain limited. This study aims to compare the performance of the K-Nearest Neighbor (KNN) and Decision Tree algorithms in predicting students' academic achievement using the publicly available Student Performance Dataset from Kaggle. A quantitative experimental approach was employed by applying data preprocessing, feature scaling for KNN, and an 80:20 train–test split. Model performance was evaluated using Accuracy, Precision, Recall, and F1-Score metrics. The experimental results demonstrate that the Decision Tree algorithm significantly outperformed KNN, achieving an accuracy of 91.65% and a weighted F1-Score of 91.44%, whereas KNN obtained an accuracy of 62.00% and a weighted F1-Score of 60.74%. The superior performance of Decision Tree indicates its ability to model complex relationships among academic and non-academic variables while being more robust to class imbalance. These findings provide empirical evidence that Decision Tree is a more suitable algorithm for developing an interpretable and reliable early warning system to support academic decision-making and student performance monitoring in educational institutions.*

## 1. Introduction

Academic achievement is one of the key measures of the success of the educational process because it reflects the level of competency mastery that students have attained during their learning. The ability to predict academic achievement early on is crucial as a basis for identifying students who are at risk of declining performance or who require prompt and appropriate academic intervention [1]. Schools can use these prediction results to design learning strategies [2], providing academic support, and improving the effectiveness of decision-making in educational management [3].

With advances in information technology, educational institutions now generate large amounts of academic and non-academic data stored in school information systems. This data includes not only academic grades but also demographic information, attendance rates, learning patterns, socioeconomic conditions, and family support. Using the Educational Data Mining (EDM) approach, this data can be processed to yield new insights that aid in data-driven prediction and decision-making. Various classification algorithms in machine learning have been widely applied in EDM to predict student performance, such as Decision Tree, Naïve Bayes, Support Vector Machine, Random Forest, and K-Nearest Neighbor (KNN). [4], [5].

Various previous studies have indicated that classification algorithms exhibit varying levels of performance on educational datasets. According to Halder et al. [4], K-Nearest Neighbor is a simple algorithm with adequate classification capabilities, but it is highly sensitive to the choice of k, the data

©2026. The Author(s). This is an open acces article under cc-by-sa

distribution pattern, and class imbalance. On the other hand, Khairy et al. [5] found that decision tree-based methods can achieve high accuracy in predicting student learning outcomes because they are able to model nonlinear relationships between academic and non-academic attributes more efficiently. Other studies have also shown that decision trees have the advantage of generating decision rules that are easy to understand, making them easier to implement in decision support systems in educational settings [6].

However, most previous studies still have several limitations. First, most studies evaluated only a single classification algorithm, thus failing to provide a clear comparison of the strengths and weaknesses of each method. Second, many studies still used local datasets with a relatively small number of features, making it impossible to capture the complexity of the factors influencing students' academic achievement. Third, discussions regarding the impact of class imbalance on the performance of classification algorithms—particularly KNN and Decision Trees—remain limited to the field of Educational Data Mining.

Within the established context, this study fills a research gap by comparing the performance of the K-Nearest Neighbor (KNN) algorithm [7] and Decision Tree using the public Student Performance Dataset from Kaggle [8]. The dataset contains a combination of academic and non-academic variables, such as demographic profiles, duration of study, attendance rates, parental support, extracurricular activities, and students' socioeconomic backgrounds. Evaluation was conducted using the metrics Accuracy, Precision, Recall, and F1-Score to objectively assess the performance of both methods.

This study makes a significant contribution by conducting a comprehensive evaluation of two popular classification algorithms using a multivariate public education dataset with an imbalanced class distribution. In addition to measuring model accuracy, this study also examines the impact of dataset characteristics on the performance of each algorithm, thereby providing a scientific basis for selecting the most appropriate classification method in the development of early warning systems for educational institutions. Based on this, the objective of this study is to compare the performance of the K-Nearest Neighbor (KNN) and Decision Tree algorithms in predicting student academic achievement through an Educational Data Mining approach, in order to identify the algorithm that yields the best accuracy, precision, recall, and F1-score on the public dataset used.

## 2. Research Methodology

### 2.1 Research Design

This study uses an experimental quantitative method with a comparative design to evaluate and compare the performance of two classification algorithms, namely K-Nearest Neighbor (KNN) [9] and Decision Trees, to predict students' academic performance using the Educational Data Mining (EDM) approach. The experimental method was chosen because it allows for the objective development, training, testing, and evaluation of models using the same dataset and consistent evaluation metrics, thereby ensuring that comparisons of results are scientifically sound. All stages of the analysis were performed using the Python programming language, with support from the Pandas library for data manipulation, NumPy for numerical calculations, Scikit-Learn for implementing machine learning algorithms and model evaluation, and Matplotlib for visualizing the research results [10].

### 2.2 Dataset Description

The study utilized the Student Performance Dataset, which is publicly available on the Kaggle platform [11]. This dataset was selected because it provides fairly comprehensive academic and non-academic information, making it suitable for research in the field of Educational Data Mining. The dataset consists of 2,392 student records with 15 attributes, including demographic characteristics, study habits, socioeconomic status, family support, school activities, and students' academic achievements. The target variable in this study is GradeClass, while the StudentID attribute was removed because it serves only as a unique identifier and does not influence the classification process. The characteristics of the dataset are shown in Table 1.

Joinforcs: Journal of informatics and information system
Volume 1, (4) 2026, p. 139-149
E-ISSN : XXXX-XXXX
©2026. The Author(s). This is an open acces article under cc-by-sa

Table 1. Dataset Characteristics

| Characteristics | Description |
|---|---|
| 0.0 (Highly Achieving) | 0.33 |
| 1.0 (High Achiever) | 0.28 |
| 2.0 (Fair) | 0.38 |
| 3.0 (Poor) | 0.42 |
| 4.0 (Fail) | 0.84 |
| Macro Avg | 0.45 |
| Weighted Avg | 0.60 |

This dataset was selected because it has more complex characteristics than most previous studies; specifically, it integrates academic and non-academic variables into a single predictive model, thereby enabling a more comprehensive evaluation of the algorithm.

### 2.3 Data Preprocessing

The preprocessing stage is carried out to improve data quality prior to the modeling process so that the classification model can produce more accurate predictions [12]. The preprocessing stage consists of the following steps:

1. Data Cleaning: The dataset was checked for structural consistency, the data types of each attribute, and the presence of duplicate or irrelevant data. The StudentID variable was removed because it did not contribute to the classification process.
2. Feature Selection: All attributes other than StudentID were used as predictor variables because, based on previous research, each is associated with factors that influence students' academic performance.
3. Feature Scaling: Standardization is performed using the StandardScaler method specific to the K-Nearest Neighbor algorithm. This step is necessary because KNN uses Euclidean distance calculations. [13] Consequently, differences in the range of values among attributes can lead to bias in the determination of nearest neighbors.

In contrast, the Decision Tree algorithm does not require standardization because its tree-building mechanism is based on attribute splitting according to impurity values, rather than distance calculations. The preprocessing steps aim to produce a more representative dataset so that the model's performance can be evaluated objectively.

### 2.4 Data Partitioning

The dataset that has undergone preprocessing is then divided into a training set and a testing set. [14] using the holdout validation method with an 80%:20% split. Eighty percent of the data was used to build the classification model, while the remaining 20% was used to evaluate the model's ability to generalize to data it had not previously been trained on.

The data was split randomly using the `train_test_split()` function in the Scikit-Learn library with the parameter `random_state = 42` so that the research results could be replicated by other researchers. The 80:20 split was chosen because it is widely used in machine learning research and is considered to provide a balance between the model training process and performance evaluation.

### 2.5 Model Development

The study developed two classification models using the same dataset, so the differences in performance observed are entirely attributable to the characteristics of the algorithms.

**Joinforcs: Journal of informatics and information system**
**Volume 1, (4) 2026, p. 139-149**
**E-ISSN : XXXX-XXXX**
**©2026. The Author(s). This is an open acces article under cc-by-sa**

1. K-Nearest Neighbor (KNN)

The first model uses the K-Nearest Neighbor (KNN) algorithm, implemented using the KNeighborsClassifier class in the Scikit-Learn library. The parameters used are:

1. n_neighbors = 5
2. metric = euclidean
3. weights = uniform

The KNN algorithm classifies an object based on the classes of its nearest neighbors using the Euclidean distance, which is defined as follows:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Description:

$x_1$ = Data Training
$x_2$ = Data Testing
$n$ = Number of Attributes

The value of k = 5 was chosen because it strikes a balance between sensitivity to local data and the model's generalization ability, based on recommendations from previous research.

a. Decision Tree

The second model uses the Decision Tree algorithm via the DecisionTreeClassifier class in the Scikit-Learn library. The parameters used include:

- criterion = gini
- max_depth = 5
- random_state = 42

The Decision Tree builds a decision tree structure by selecting the attribute with the lowest impurity value based on the Gini Index, so that each data splitting process produces increasingly homogeneous nodes. The constraint `max_depth = 5` is applied to reduce the risk of overfitting without compromising the model's ability to identify classification patterns.

In addition to generating highly accurate predictions, Decision Trees can also produce rules that are easy to interpret, making them suitable for use as a basis for decision-making in educational institutions.

**2.6 Model Evaluation**

The performance of the two algorithms was evaluated using a confusion matrix [15] which illustrates the relationship between the model's predicted results and the actual labels.

1. Accuracy : measuring the proportion of correct predictions relative to the entire test dataset.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Precision : Precision indicates the model's accuracy when making positive predictions.

$$Precision = \frac{TP}{TP+FP}$$

3. Recall : Recall measures a model's ability to identify all data points that actually belong to the positive class.

**Joinforcs: Journal of informatics and information system**
**Volume 1, (4) 2026, p. 139-149**
**E-ISSN : XXXX-XXXX**
**©2026. The Author(s). This is an open acces article under cc-by-sa**

$$Recall = \frac{TP}{TP + FN}$$

4. F1-Score : F1-Score is the harmonic mean of Precision and Recall, making it capable of providing a more representative evaluation for datasets with an imbalanced class distribution.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

These four metrics were selected because they complement each other in evaluating the performance of classification models, particularly in cases of multi-class classification characterized by class imbalance. The research stages are shown in Figure 1 below :

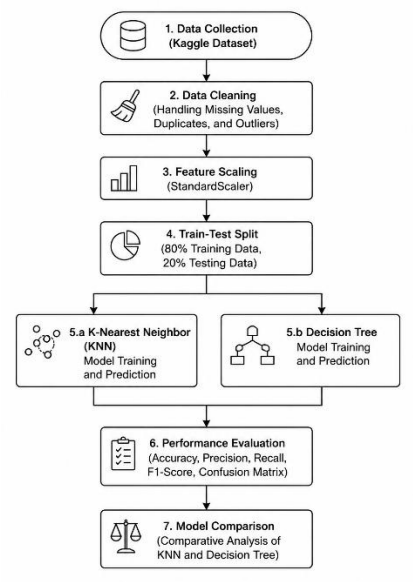



Figure 1. *Flowchart* of the research method

## 3. Results and Discussion

### 3.1 Performance Evaluation of the K-Nearest Neighbor Algorithm

Before evaluating the model, the class distribution in the Student Performance Dataset was analyzed to determine the balance of sample sizes across each performance category. The results of the analysis are shown in Figure 2.

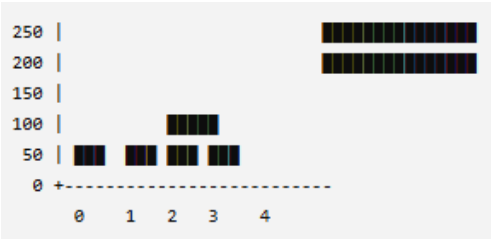



Figure 2. Distribution of GradeClass

Figure 2 shows that the data exhibits class imbalance, with GradeClass 4 (Fail) having the largest number of samples compared to the other classes. Conversely, GradeClass 0 (High Achiever) has the fewest samples. This imbalance can affect the performance of classification algorithms, particularly distance-based

**Joinforcs: Journal of informatics and information system**
**Volume 1, (4) 2026, p. 139-149**
**E-ISSN : XXXX-XXXX**
**©2026. The Author(s). This is an open acces article under cc-by-sa**

ones such as K-Nearest Neighbor (KNN), because the process of determining the nearest neighbors tends to be dominated by the class with more samples.

The K-Nearest Neighbor (KNN) model was trained using 80% of the training data and tested using 20% of the test data, consisting of 479 samples. After feature standardization using StandardScaler, the model achieved an accuracy of 62.00% with a weighted F1-score of 60.74%. Detailed evaluation results are shown in Table 2.

Table 2. Classification Report Algoritma KNN

| Honors Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0.0 (Highly Proficient) | 0.33 | 0.14 | 0.19 | 22 |
| 1.0 (High Achiever) | 0.28 | 0.22 | 0.25 | 49 |
| 2.0 (Fair) | 0.38 | 0.46 | 0.41 | 85 |
| 3.0 (Poor) | 0.42 | 0.36 | 0.39 | 86 |
| 4.0 (Fail) | 0.84 | 0.90 | 0.87 | 237 |
| Macro Avg | 0.45 | 0.42 | 0.42 | 479 |
| Weighted Avg | 0.60 | 0.62 | 0.61 | 479 |

Based on Table 2, the KNN model's performance shows significant differences across classes. The model is able to classify the GradeClass 4 (Fail) class very well, as indicated by a precision of 0.84, a recall of 0.90, and an F1-score of 0.87. In contrast, performance on minority classes is relatively low. The GradeClass 0 (High Achiever) class achieved a recall of only 0.14, indicating that the model failed to identify the majority of students in the high-achiever category.

To provide a more detailed picture of the model's prediction patterns, the classification results are visualized using a confusion matrix, as shown in Figure 3.

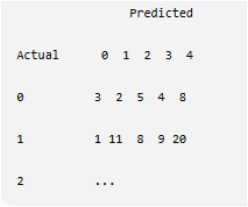



Figure 3. Confusion Matrix of KNN

Figure 3 shows that the majority of classification errors occur in the minority classes, where many data points in GradeClass 0 and 1 are incorrectly predicted as belonging to the class with more samples. This pattern indicates that the KNN algorithm tends to be biased toward the majority class due to its classification mechanism, which relies on comparing Euclidean distances between samples. Although a standardization process was included, the dominance of the majority class in terms of data volume causes the nearest-neighbor search to more frequently point to that class, thereby reducing the model's ability to identify the minority class.

These results are consistent with the study by Halder et al. [4] which states that the performance of KNN is greatly influenced by the data distribution, the choice of the value of k, and the presence of class imbalance. In datasets with an uneven class distribution, the KNN algorithm tends to be biased toward the majority class, thereby reducing the precision, recall, and F1-score for classes with few samples. Therefore, this study demonstrates that KNN is not suitable for direct application to educational datasets characterized by class imbalance without the support of data balancing techniques, such as SMOTE or class weighting..

**Joinforcs: Journal of informatics and information system**
**Volume 1, (4) 2026, p. 139-149**
**E-ISSN : XXXX-XXXX**
**©2026. The Author(s). This is an open acces article under cc-by-sa**

### 3.2 Performance Evaluation of the Decision Tree Algorithm

The second model was built using the Decision Tree algorithm with the parameters criterion = gini and max_depth = 5. Based on the results of testing using the same dataset, the Decision Tree achieved an accuracy of 91.65% with a weighted F1-score of 91.44%. The evaluation results are presented in Table 3.

Table 3. Classification Report Algoritma Decision Tree

| Honors Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0.0 (Highly Proficient) | 0.76 | 0.59 | 0.67 | 22 |
| 1.0 (High Achiever) | 0.88 | 0.88 | 0.88 | 49 |
| 2.0 (Fair) | 0.94 | 0.87 | 0.90 | 85 |
| 3.0 (Poor) | 0.89 | 0.87 | 0.88 | 86 |
| 4.0 (Fail) | 0.94 | 0.99 | 0.96 | 237 |
| Macro Avg | 0.88 | 0.84 | 0.86 | 479 |
| Weighted Avg | 0.91 | 0.92 | 0.91 | 479 |

Based on Table 3, the Decision Tree algorithm demonstrated more consistent performance across all classes compared to KNN. The precision, recall, and F1-Score values were in the high range for nearly all student achievement categories. In GradeClass 4 (Failed), the model achieved a precision of 0.94, a recall of 0.99, and an F1-Score of 0.96, demonstrating excellent ability in identifying the majority class. Meanwhile, for minority classes such as GradeClass 1 (High Achiever), the model still maintained a precision and recall of 0.88, while GradeClass 0 (Top Achiever) achieved an F1-Score of 0.67, which is significantly better than the KNN results.

To provide an overview of the distribution of prediction results across each class, the confusion matrix for the Decision Tree algorithm is shown in Figure 4.

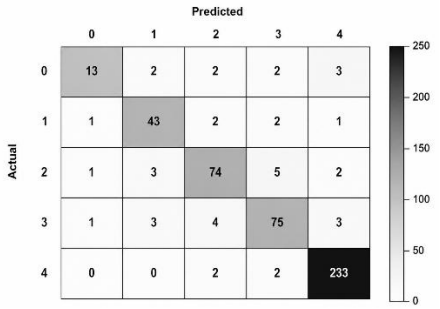



Figure 4. Confusion Matrix of Decision Tree

As shown in Figure 4, most of the prediction results fall along the main diagonal of the confusion matrix, indicating that the majority of the data was successfully classified according to its true class. Classification errors occurred only in a small portion of the samples, particularly in classes with relatively few data points. This pattern suggests that the Decision Tree performs better at distinguishing characteristics between classes, even in data with an imbalanced distribution.

In addition to delivering high classification performance, Decision Trees also allow for the identification of the attributes that have the greatest influence on the prediction process through feature importance analysis, as shown in Figure 5.

**Joinforcs: Journal of informatics and information system**
**Volume 1, (4) 2026, p. 139-149**
**E-ISSN : XXXX-XXXX**
**©2026. The Author(s). This is an open acces article under cc-by-sa**

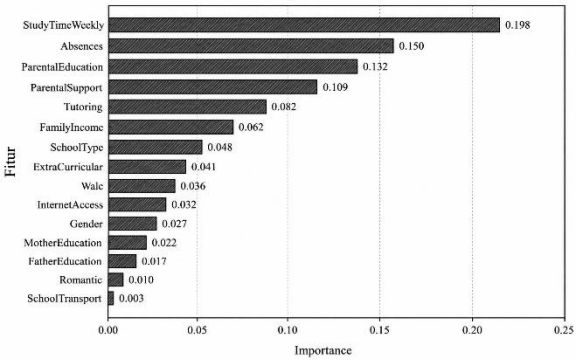


Figure 5. Feature Importance of Decision Tree

Figure 5 shows that the variables StudyTimeWeekly, Absences, ParentalEducation, ParentalSupport, and Tutoring are the factors that play the most significant role in distinguishing students' academic achievement. These findings indicate that academic achievement is influenced not only by academic factors themselves but also by study habits and family support. This information underscores the advantages of the Decision Tree as a model that is not only capable of making highly accurate predictions but also provides an interpretation of the factors influencing the prediction results.

The high performance of the Decision Tree indicates that this algorithm is capable of building effective classification rules based on a combination of academic and non-academic attributes. Unlike KNN, which relies on the proximity of data points, the Decision Tree performs data splitting in stages using the Gini Index, making it more adaptive in modeling nonlinear relationships between variables. This mechanism allows the model to maintain stable performance even when the dataset exhibits class imbalance.

The results of this study are consistent with the findings of Khairy et al. [5] which confirms that decision-tree-based algorithms are highly effective in modeling the complex relationships among factors that influence students' academic achievement. Another study by Jijo and Abdulazeez [6] It adds that decision trees can produce models that are easy to understand through rule extraction, thereby not only achieving high accuracy but also enhancing transparency in the decision-making process. Thus, these findings confirm that decision trees are the most appropriate algorithm to use as the foundation for developing early warning systems in educational institutions, as they combine high accuracy with model interpretability.

### 3.3 Comparative Analysis of KNN and Decision Tree

A comprehensive comparison of the performance of the K-Nearest Neighbor (KNN) and Decision Tree algorithms based on evaluation metrics is presented in Table 4, while a visual comparison of the performance of the two models is shown in Figure 6.

Table 4. Summary of Model Performance Comparison

| Algorithm | Accuracy | Weighted Precision | Weighted Recall | Weighted F1-Score |
|---|---|---|---|---|
| K-Nearest Neighbor (KNN) | 62.00% | 60.00% | 62.00% | 60.74% |
| Decision Tree | 91.65% | 91.00% | 92.00% | 91.44% |

**Joinforcs: Journal of informatics and information system**
**Volume 1, (4) 2026, p. 139-149**
**E-ISSN : XXXX-XXXX**
**©2026. The Author(s). This is an open acces article under cc-by-sa**

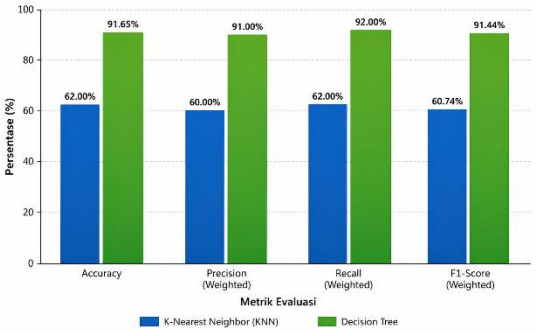



Figure 6. Comparison of KNN and Decision Tree Performance

Based on Table 4 and Figure 6, the Decision Tree algorithm consistently outperformed the K-Nearest Neighbor (KNN) algorithm across all evaluation metrics. The Decision Tree achieved an accuracy of 91.65%, while KNN reached only 62.00%. Similar differences were also observed in the weighted Precision, weighted Recall, and weighted F1-Score values, indicating that the Decision Tree algorithm exhibits more stable classification performance across all student achievement categories.

These performance variations indicate that the nature of the algorithm strongly influences the model's ability to process educational datasets that include both academic and non-academic attributes and exhibit class imbalance. In this study, the Decision Tree builds the model by hierarchically partitioning the data using the Gini Index, so that each node produces a more homogeneous dataset. This method helps the model extract complex relationships between features without being affected by the scale of the data or the dominance of a particular class.

In contrast, the KNN algorithm classifies data based on the closest distance using Euclidean distance, making it highly dependent on the distribution pattern of the data in the feature space. If the number of samples in the majority class is significantly greater than that of other classes, the nearest-neighbor search tends to produce predictions that are biased toward the majority class. This condition makes the model less capable of identifying the minority class, as indicated by the low recall and F1-score values for the GradeClass 0 and GradeClass 1 classes.

In addition to improving accuracy, decision trees also offer the advantage of interpretability. The structure of a decision tree makes it easy to explain each classification process through decision rules that are easy to understand. This advantage adds value to academic performance prediction systems because the prediction results not only indicate students' performance categories but can also serve as a basis for teachers or school administrators to design learning strategies or academic interventions.

The results of this study are consistent with the study by Jijo and Abdulazeez [6], which states that decision trees have good classification capabilities for multivariate data and produce models that are easy to interpret. On the other hand, the findings of this study also reinforce the results of the study by Halder et al. [4], which explains that KNN performance tends to decline when applied to datasets with an imbalanced class distribution. Therefore, based on the results of the experiments conducted, Decision Trees can be recommended as a more suitable algorithm for developing a student academic performance prediction system based on Educational Data Mining, especially when model interpretability and performance stability are key considerations.

### 3.4 Discussion

The research findings confirm that the Decision Tree is the best algorithm for predicting students' academic performance in this dataset. With an accuracy of 91.65%, this model was more effective than KNN at analyzing the patterns of relationships between academic and non-academic variables.

The performance of the two algorithms differs due to the way their models are built. KNN relies on measuring the distance between data points, so it is highly influenced by the data distribution and the presence of class imbalance. Although the data was standardized using StandardScaler, the dominance of the majority class still causes the nearest-neighbor search process to be biased. This is evident from the low recall value for the high-performance class, which reached only 0.14.

**©2026. The Author(s). This is an open acces article under cc-by-sa**

In contrast, Decision Trees build models by dividing attributes in a hierarchical manner based on impurity values, allowing them to capture complex relationships between variables without being affected by the scale of the data. Attributes such as StudyTimeWeekly, Absences, ParentalEducation, ParentalSupport, and socioeconomic factors can be combined into decision rules that more accurately reflect real-world conditions.

Based on its implementation, this study indicates that the Decision Tree model has greater potential to serve as an early warning system in educational settings. In addition to demonstrating high accuracy, the model provides classification rules that are easy for teachers and school administrators to understand. Therefore, the predictions generated are not merely academic data but can also serve as a reference for formulating mentoring strategies, providing academic guidance, and developing intervention programs for students at risk of declining academic performance.

This study has several limitations. The evaluation was conducted using only one public dataset split in an 80:20 ratio, so the findings cannot yet be generalized to the full range of educational data. Furthermore, the study did not apply class imbalance mitigation techniques such as SMOTE or k-fold cross-validation, which could potentially enhance the model's stability. Therefore, further studies are recommended to test other algorithms—such as Random Forest, XGBoost, or Support Vector Machine—and to implement data balancing and cross-validation techniques to ensure more comprehensive results.

## 4. Conclusion

This study successfully conducted a comparative analysis of the K-Nearest Neighbor (KNN) and Decision Tree algorithms in predicting student academic performance using the Educational Data Mining (EDM) approach, utilizing the Student Performance Dataset obtained from Kaggle. Based on the evaluation results using the metrics Accuracy, Precision, Recall, and F1-Score, the Decision Tree algorithm consistently outperformed KNN, with an accuracy of 91.65% and a weighted F1-Score of 91.44%, while KNN achieved only 62.00% accuracy and a weighted F1-Score of 60.74%. This difference in performance indicates that the Decision Tree is more effective at handling the characteristics of a dataset that combines academic and non-academic attributes and exhibits class imbalance.

The contribution of this study lies in presenting a comparative evaluation of the KNN and Decision Tree algorithms using a public dataset that integrates various academic and non-academic factors in the field of Educational Data Mining. The results of this study provide an empirical reference for researchers and practitioners in selecting the most appropriate classification algorithm for developing academic performance prediction systems. From an implementation perspective, the Decision Tree model has the potential to serve as an early warning system in educational institutions because, in addition to achieving high accuracy, it can also generate clear decision rules (rule extraction) that are easily understood by both teachers and school administrators.

This study has several limitations. The evaluation used only one public dataset, with 80 % of the data allocated for training and 20 % for testing; therefore, the findings cannot yet be generalized to all types of educational data. Furthermore, the study has not yet implemented techniques for handling class imbalance or cross-validation methods that could improve the stability of the model's performance. Therefore, further studies are recommended to test other classification algorithms such as Random Forest, XGBoost, or SVM, as well as to incorporate data balancing techniques like SMOTE and the Stratified k-Fold Cross-Validation approach in order to obtain a more robust predictive model and better generalization capabilities.

## REFERENCES

[1] M. Anderson, J. Williams, R. Thompson, E. Johnson, and C. James, "Machine Learning Applications in Student Performance Prediction."

[2] H. Lestari, A. Irma Purnamasari, and T. Suprapti, "Penerapan Data Mining Menggunakan Algoritma C4.5 Untuk Prediksi Prestasi Belajar Siswa Di Mts Yamuallim Panongan," 2024. doi: https://doi.org/10.36040/jati.v8i2.8312

**©2026. The Author(s). This is an open acces article under cc-by-sa**

[3] H. Lestari, A. Irma Purnamasari, and T. Suprapti, "Penerapan Data Mining Menggunakan Algoritma C4.5 Untuk Prediksi Prestasi Belajar Siswa Di Mts Yamuallim Panongan," 2024. doi: https://doi.org/10.36040/jati.v8i2.8312

[4] R. K. Halder, M. N. Uddin, M. A. Uddin, S. Aryal, and A. Khraisat, "Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications," *J. Big Data*, vol. 11, no. 1, Dec. 2024, doi: https://doi.org/10.1186/s40537-024-00973-y.

[5] D. Khairy, N. Alharbi, M. A. Amasha, M. F. Areed, S. Alkhalaf, and R. A. Abougalala, "Prediction of student exam performance using data mining classification algorithms," *Educ. Inf. Technol. (Dordr).*, vol. 29, no. 16, pp. 21621–21645, Nov. 2024, doi: https://doi.org/10.1007/s10639-024-12619-w.

[6] B. T. Jijo and A. M. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 1, pp. 20–28, Jan. 2021, doi: https://doi.org/10.38094/jastt20165.

[7] M. I. Mubarok, C. Informatika, S. Karawang, J. H. Ronggo Waluyo, T. Timur, and I. Karawang, "Penerapan Algoritma K-Nearest Neighbor (Knn) Dalam Klasifikasi Penilaian Jawaban Ujisan Esai," 2023.

[8] N. Situmeang, E. Artamezia Komarsyah, A. Fauzi, and J. Sagala, "Penggunaan Algoritma K-Nearest Neighbor (KNN) Untuk Mendeteksi Stunting Pada Anak," *Jepri Sagala INNOVATIVE: Journal Of Social Science Research*, vol. 5, pp. 8401–8417, 2024.

[9] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, "Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction," *Sci. Rep.*, vol. 12, no. 1, Dec. 2022, doi: https://doi.org/10.1038/s41598-022-10358-x.

[10] I. Harouni, "The Modern Methods of Data Analysis in Social Research," vol. 6, no. 1, pp. 56–70, Mar. 2024, doi: https://doi.org/10.34118/sej.v6i1.3806.

[11] J. Homepage *et al.*, "MALCOM: Indonesian Journal of Machine Learning and Computer Science Implementation of Decision Tree Algorithm and Support Vector Machine for Lung Cancer Classification Implementasi Algoritma Decision Tree dan Support Vector Machine untuk Klasifikasi Penyakit Kanker Paru," vol. 3, pp. 15–19, 2023. doi: https://doi.org/10.57152/malcom.v3i1.591

[12] Q. A. Puteri, T. Sagirani, and J. Lemantara, "Perbandingan Algoritma Naïve Bayes dan K-Nearest Neighbor (KNN) untuk Mengetahui Keakuratan Diagnosa Penyakit Diabetes," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 9, no. 3, pp. 247–254, Dec. 2023, doi: https://doi.org/10.25077/teknosi.v9i3.2023.247-254.

[13] K. Kualitas *et al.*, "Klasifikasi Kualitas Jambu Air Madu Deli Merah, Dengan Menerapkan Metode K-NN model Euclidean Distance," vol. 07, 2025, doi: https://doi.org/10.54209/jatilima.v7i05.1633.

[14] A. S. Fauzan, A. Irma, P. Sari, and I. Ali, "Analisis Perbandingan Algoritma Decisioin Tree Dan Naïve Untuk Mengevaluasi Prestasi Belajar Siswa Studi Kasus: Smk Al-Musyawirin," 2024.

[15] I. Habib Kusuma and N. Cahyono, "Analisis Sentimen Masyarakat Terhadap Penggunaan E-Commerce Menggunakan Algoritma K-Nearest Neighbor," vol. 8, no. 3, 2023. doi: https://doi.org/10.30591/jpit.v8i3.5734