A Comparison Study of Classification Algorithms for Data Mining

OZER OZDEMIR, BERFIN ESEN, EGEMEN AKDEMIR
Department of Statistics,
Eskisehir Technical University,
İki Eylül Campus, 26555,
TÜRKİYE

Abstract: - Data mining is the process of identifying patterns, correlations, and anomalies in large data sets. Data mining involves preparing selected data sets appropriately for analysis, thus revealing expected or unexpected meaningful information. The aim of the classification method is to create a model that can predict which class or category a new data sample belongs to based on its characteristics. In our study, our dataset is a dataset obtained from the UCI Machine Learning website that examines Student Dropout and Academic Performance. This dataset focuses on various attributes such as marital status, gender, and socio-economic factors, etc. In this study, topics such as data preparation for data mining and model performance evaluation criteria are discussed. In addition, decision trees and algorithms such as XGBoost, Catboost, Naive Bayes, and AdaBoost are also examined. The data set employed here is a large collection that investigates student dropout and academic achievement. During the application phase, two separate models were compared, each trained on different splits of 80 percent and 75 percent of the data. Statistics from algorithms like Decision Trees, Random Forest, Naive Bayes, Gradient Boosting, and CatBoost were compared, and the best accuracy, 78.90 percent, was achieved by the CatBoost algorithm trained on the 75 percent split. This result led to the conclusion that the method built with CatBoost is the most effective classification approach for this problem.

Key-Words: - Data mining, Classification algorithms, CatBoost, Gradient Boosting, XGBoost, AdaBoost.

Received: June 14, 2025. Revised: March 18, 2026. Accepted: May 11, 2026. Published: July 13, 2026.

1 Introduction

Data mining is an approach that involves an intermixture of techniques to identify a block of data or decision-making knowledge in the database and eradicating these data in such a way that they can be used in decision support, forecasting, and estimation, [1]. The data is often voluminous, but it has useful. Two major preferred models that can be created in data mining are predictive and descriptive. Under these two models, there are various tasks that are used in the data mining process. On the basis of various historical data, a predictive model makes an estimation about the values of data using recognized results found from various data. On the other hand, the descriptive model identifies patterns or relationships in data. Unlike the predictive model, a descriptive model serves as a way to explore the properties of the data observed, not to predict new properties, [2].

Although it has only recently become a hot topic, the techniques used in data mining have been around since the 1980s. This emerging technology combines statistical analysis, machine learning, and database management to extract information from large database systems.

Several definitions of data mining have been laid forth by various researchers. Some have defined data mining as a process of discovering useful or actionable knowledge in large-scale data, [3], [4], [5], [6], [7], [8]. According to [9], data mining is the process of discovering insightful, interesting, and novel patterns, as well as descriptive, understandable, and predictive models from large-scale data. Another definition of data mining, as coined by [4] and [10], is the extraction of interesting (non-trivial, implicit, previously unknown, and potentially useful) patterns or knowledge from huge amounts of data. Data mining also means knowledge discovery from data, which describes the typical process of extracting useful information from raw data, [3], [4], [5].

As pointed out by [11], many people treat data mining as a synonym for another popularly used term, Knowledge Discovery in Databases, or KDD. This observation is quite true if one can closely look at the interpretations which have been made by several researchers, such as [1], [2], [5], [12] about

data mining. However, as pointed out by [11], data mining is also treated simply as an essential step in the process of knowledge discovery in databases.

[13] studied the dropout likelihood of Electrical Engineering (EE) students after their first semester or before starting the program, along with success factors. Decision trees achieved accuracy rates of 75-80%. The benefits of cost-sensitive learning and detailed misclassification analysis were demonstrated, along with ways to improve predictions without collecting additional student data [13].

[14] studied this project. The decision tree technique, one of the data mining techniques, was applied using the annual financial indicators of 173 companies operating in the industrial and service sectors listed in the MKB 100 index for the years 2004–2006. The most important variables distinguishing companies operating in the industrial and service sectors based on the selected financial indicators were identified, [14].

[15] aims to identify student dropout patterns at the University of Nariño (Colombia) using socioeconomic, academic, disciplinary, and institutional data.

Between 2004 and 2006, researchers examined data from three different groups of students. They used decision tree classification methods to identify clear patterns related to student dropouts. Among the tested models, the J48 algorithm showed the highest accuracy. The findings from this study provide useful information for creating better strategies and policies to improve student retention, [15].

[16] examined the issue of early school dropout by introducing a new method and a custom classification algorithm to predict student retention early on. The researchers used data from 419 high school students in Mexico. They ran experiments to find the best dropout indicators at different points in the course and compared their algorithm to traditional classification methods. The results showed that the proposed algorithm could accurately predict dropout risk within the first 4 to 6 weeks of the program. This model can act as an early warning system, enabling timely actions to lower student dropout rates, [16].

[17] studied success prediction in crowdfunding platforms, where the average project success rate is about 40%. Most platforms use an "all-or-nothing" funding model. Predicting project outcomes is crucial. In this study, project success was viewed as a survival analysis issue using censored regression techniques. By examining data from Kickstarter projects and related tweets, the researchers discovered that models including both successful and unsuccessful projects performed better than those trained only on successful ones. Additionally, adding early-stage time-based features further improved the models' accuracy, [17].

[18] studied the main factors that predict the success of Kickstarter campaigns. They used feature engineering and different data mining methods, including Logistic Regression, SVM, Bagging, and Boosting. Cross-validation results showed that bagging and gradient boosting were the most reliable predictive methods. Future research should look into Bayesian semi-parametric approaches, [18].

[19] studied the Multivariate Adaptive Regression Splines (MARS) algorithm to model the mood state correction factor for thermal sensation. Mood state is seen as a thermal sensation parameter, along with environmental factors. Using data collected from a university study hall, the results from the MARS model are compared to the black-box model. The observed correlation was high at 94.2%. This shows that the data mining algorithm models the mood state correction factor well, supporting the black-box model, [19].

K-means clustering is a widely used unsupervised learning algorithm that partitions data into K distinct clusters by minimizing within-cluster variance. Each cluster is represented by its centroid, and the algorithm iteratively updates cluster assignments until convergence. Due to its simplicity and efficiency, it is commonly applied in data mining and pattern recognition tasks, [20].

The Student Dropout dataset is designed to analyze the factors leading to early school dropout and to develop early interventions to prevent this issue by identifying at-risk students. The findings can enable the development of early warning systems to identify risk factors like poor performance or absenteeism and create tailored guidance and support programs for these students.

2 Problem Formulation

2.1 Decision Trees Algorithms

2.1.1 ID3

This algorithm is considered a very simple decision tree algorithm. ID3 uses information gain as the splitting criterion. The tree growth stops when all examples belong to a single value of the target attribute or when the best information gain is not greater than zero. ID3 does not apply any pruning procedures and does not handle numerical attributes or missing values, [21].

It is a method that works only with categorical data. In the first step of each iteration, the entropy of the vector containing the class information of the data samples is determined. Entropy is the maximization of uncertainty, [22]. First, the entropy of the dataset is computed. Then, the class-dependent entropies corresponding to each feature are calculated and subtracted from the initial entropy. The resulting value represents the information gain associated with that feature. The feature with the highest information gain is selected to determine the branching of the decision tree at that stage, [23].

The basic strategy used by ID3 is to choose splitting attributes with the highest information gain first. The amount of information associated with an attribute value is related to the probability of occurrence, [22].

2.1.2 C4.5 and C5.0

The C4.5 algorithm provides several advantages over the ID3 algorithm in the following aspects. While constructing the decision tree, missing data is not considered. In other words, the gain ratio is calculated using only those records that have complete data. The C4.5 algorithm predicts missing values using other available data and variables and includes these predictions in the gain ratio calculation, [22]. As a result, the tree generated is more sensitive and capable of producing more meaningful rules.

2.1.3 CART

The aim of the CART algorithm is not merely to produce a single tree, but to generate a sequence of nested, pruned trees, each being an optimal candidate. The "appropriately sized" or "honest" tree is determined by evaluating the prediction performance of each tree in the pruning sequence using independent test data. Unlike C4.5, CART does not rely on internal performance metrics based on the training data for tree selection. Instead, tree performance is always measured using either independent test data or cross-validation, and the selection of the final tree proceeds only after such evaluation, [24].

2.1.4 CHAID

The CHAID (Chi-square Automatic Interaction Detector) algorithm was first developed by [25]. This method aims to group data in such a way that the dependent variable is best explained by the independent variables. CHAID is widely used for classification problems and regression analysis, although it is particularly preferred for classification tasks, [25].

2.2 Naive Bayes

The Naive Bayes algorithm is a simple, yet powerful probabilistic classification method based on Bayes' theorem, assuming that features are conditionally independent given the class label, [26]. Although this "naive" independence assumption rarely holds perfectly in real-world data, Naive Bayes is effective and fast in many domains such as text classification, spam filtering, and medical diagnosis, [27], [28]. In academic research, Naive Bayes is widely regarded as a fundamental method due to its simplicity and interpretability, making it popular both for teaching and practical applications, [26], [27], [28], [29], [30], [31], [32], [33].

2.3 K-Nearest Neighbors (KNN)

The K-Nearest Neighbors (KNN) algorithm is a fundamental and intuitive classification method widely used in machine learning, [34]. This approach is highly effective and easy to understand in pattern recognition and classification problems. A neighborhood is typically defined using distance metrics such as Euclidean distance or Manhattan distance, and the choice of the appropriate distance metric directly affects model performance, [35].

2.4 Multi-Layer Perceptron

There are many studies in the literature related to artificial neural networks, [36], [37]. One of the architectures of these neural networks is the multilayer perceptron. The MultiLayer Perceptron (MLP) is one of the most well-known artificial neural network architectures, based on the Rosenblatt perceptron. MLPs can approximate any continuous, bounded, and differentiable nonlinear function with arbitrary precision. This is achieved through the additive composition of base functions called ridge functions. However, this theory does not specify the number of neurons needed or how to optimize the network weights.

MLPs consist of several layers of neurons arranged in parallel, where information flows only forward (feedforward structure). The first layer is the input layer, which receives the problem’s features and passes them to the hidden layers. Hidden layers transform inputs nonlinearly into another space, so nonlinear activation functions are typically used. The final layer is the output layer that produces the problem’s results.

A single output node is usually used for regression or binary classification, while multi-class classification employs one output node per class. After training, numerical outputs are mapped to classes. For binary classification, sigmoid activation is commonly applied at the output node, where

values above 0.5 predict one class, and values at or below 0.5 predict the other, [38].

2.5 Gradient Boosting

Gradient Boosting(GBM) is an algorithm that transforms weak learners into strong learners using the gradient boosting method. The Gradient Boosting algorithm is a tree-based algorithm. Understanding the working principle of decision trees is important in terms of gaining insight into Gradient Boosting. Decision trees are one of the commonly used methods for classification and regression predictions. Compared to many methods, being easy to understand and interpret is one of the advantages of this method. A decision tree starts from the root node, interprets the data, branches downward, and forms leaves, [39].

We can see that the feature importances of the gradient boosted trees are somewhat like the feature importances of the random forests, though the gradient boosting completely ignored some of the features. As both gradient boosting and random forests perform well on similar kinds of data, a common approach is to first try random forests, which work quite robustly. If random forests work well but prediction time is at a premium, or it is important to squeeze out the last percentage of accuracy from the machine learning model, moving to gradient boosting often helps, [40].

In conclusion, the Gradient Boosting algorithm is a powerful machine learning tool that can be applied in many areas. Despite having several disadvantages, GBM is a popular choice for classification and regression problems due to its high accuracy, flexibility, and robustness to noise.

2.5.1 AdaBoost

AdaBoost (Adaptive Boosting) is a powerful ensemble learning method widely used for classification problems in machine learning. Introduced by [41], this algorithm constructs a strong classifier by combining multiple weak learners, typically simple models like decision trees, through a sequential training process, [41]. The fundamental idea of AdaBoost is that each new learner focuses more on the training instances that previous learners misclassified, thereby improving the model's overall performance iteratively, [42].

2.5.2 XGBoost

XGBoost (Extreme Gradient Boosting) is a tree-based ensemble algorithm that stands out with its high accuracy and computational efficiency in supervised learning problems. Particularly effective for classification and regression tasks, this algorithm addresses many of the shortcomings of classical boosting methods and is widely preferred in modern data science applications, [43].

In academia, there are numerous studies on XGBoost. For example, the original study by [43] detailed the algorithm's technical structure and engineering optimizations. [44] examined the impact of regularization techniques on overfitting in decision trees, theoretically supporting XGBoost's advantages. Moreover, the comprehensive review by [45] on gradient boosting machines demonstrated how the algorithm performs in various applications.

2.5.3 CatBoost

CatBoost, short for Categorical Boosting, is an open-source gradient boosting decision tree (GBDT) developed by researchers at Yandex, [46]. It is built upon the GBDT framework but introduces innovative methods for handling categorical variables and mitigating overfitting.

CatBoost incorporates both conventional and unique regularization strategies to prevent overfitting. Firstly, it limits the maximum depth of trees to control model complexity. Additionally, it applies L2 regularization to leaf values, penalizing large weights, [47].

In the literature, it has been applied successfully in finance, healthcare, astronomy, cybersecurity, and bioinformatics, [48].

3 Problem Solution

In this study, the comparative performance of prediction models related to university students' success statuses, namely "Dropout," "Enrolled," and "Graduate," is examined, [49], [50]. In this context, a detailed performance analysis has been conducted using nine different machine learning algorithms: Decision Tree, Random Forest, XGBoost, K-Nearest Neighbors, Naive Bayes, Multi-layer Perceptron, AdaBoost, Gradient Boosting, and CatBoost. This analysis is based on both overall accuracy and class-based precision, recall, and F_1-score metrics.

The data source of the study consists of demographic, academic, and behavioral features obtained from 4425 student records. Demographic information such as age, gender, and department, along with academic and behavioral data such as grade point averages, credit counts, online participation rates, and forum interactions, have been transformed into a single observation row for each student, [49], [50]. By splitting the data into two different ways, 80% training, 20% testing (Mod A), and 75% training, 25% testing (Mod B), it became possible to observe how robust the models are to

small changes in training data size. For missing data points, a combination of category-specific nearest neighbor and statistical imputation methods was used, and continuous variables were normalized using Min–Max scaling. Since the "Enrolled" class contained fewer examples compared to the others, data balancing was applied in the training set using the SMOTE technique, but no positive outcome was achieved with this method.

For each algorithm, hyperparameter optimization was performed using five-fold cross-validation (5-fold CV). In decision trees, the maximum depth was limited to 10; in Random Forest, 150 trees and "sqrt" feature selection were used; in XGBoost, 200 iterations with a learning rate of 0.1 were chosen; and in CatBoost, a depth of 6 and a learning rate of 0.03 were preferred. In the deep learning-based Multi-layer Perceptron (MLP), two hidden layers were used with 64 and 32 neurons respectively, a learning rate of 0.001, and 100 epochs. In the K-Nearest Neighbors algorithm, k=7 and distance-based weighting were applied. In AdaBoost and Gradient Boosting methods, the depth of the base decision tree was kept in the range of 3–5, and a total of 100–200 trees were configured. These settings were balanced to keep model complexity under control while minimizing the risk of overfitting.

3.1 Performance Findings

3.1.1 MOD A (80% Train, 20% Test)

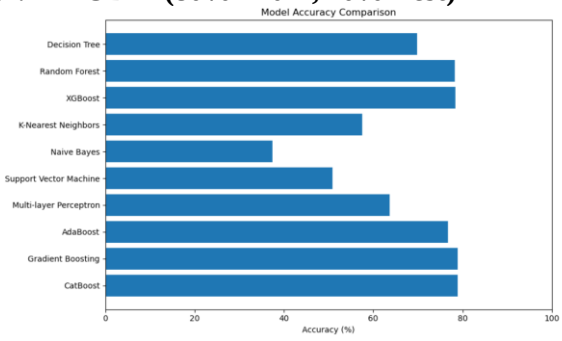

Fig. 1: Mod A(Model accuracy comparison)
Source: created by the authors.

The accuracy chart (Figure 1) shows that CatBoost and Gradient Boosting achieve the highest performance with approximately 79%, followed closely by XGBoost and Random Forest with very similar results. These four tree-based ensemble methods have a clear advantage in capturing the complex relationships in the dataset. AdaBoost lags slightly behind this group with around 77%, but still offers strong performance, while the single-tree Decision Tree remains around 70%, MLP at 64%, K-NN at 57%, and Naive Bayes, which relies on the assumption of independence between variables, performs the worst at only 37%.

In K-NN, both precision and recall remain low, reflecting the limited flexibility of the distance-based method in handling complex, high-dimensional decision surfaces.

These results indicate that tree-based ensemble models should be the first choice in student success classification problems.

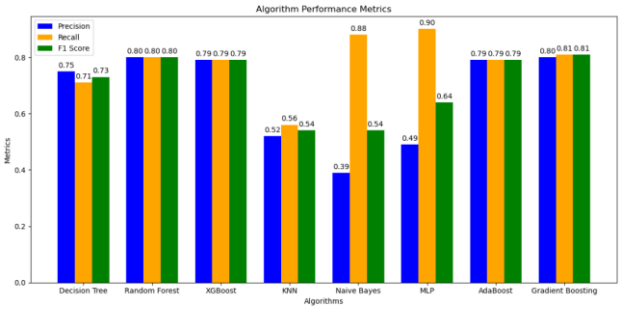

Fig. 2: Mod A(Algorithm performance metrics)
Source: created by the authors.

The algorithm performance bar chart (Figure 2) clearly reveals the strengths and weaknesses of the models across three metrics (precision, recall, and F_1 -score). The bars for Gradient Boosting and Random Forest are nearly equal in height and clustered around the ≈0.80 range, indicating that these two ensemble methods maintain a balanced approach in both correct classification rate (precision) and the ability to capture actual instances (recall); therefore, their F_1 -scores are also at the highest level. XGBoost and AdaBoost achieve a similar balance and are included in the "top-level" group.

In the Decision Tree model, precision is slightly higher than recall; the model behaves a bit more cautiously, reducing false positives while slightly increasing the false-negative rate.

3.1.2 MOD B (75% Train, 25% Test)

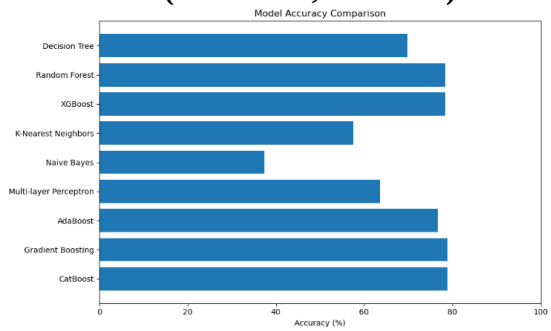

Fig. 3: Mod B(Model accuracy comparison)
Source: created by the authors.

In the 75% training – 25% test split, the overall accuracy table (Figure 3) shows that tree-based

ensemble methods, especially CatBoost (0.7890) and Gradient Boosting (0.7887), still provide the highest accuracy. These two models slightly outperform XGBoost (0.7842) and Random Forest (0.7831), forming a "leader group" of four models at almost the same level. The accuracy of the single-tree Decision Tree drops to 0.6984, revealing its limited ability to capture complex patterns. The distance-based K-Nearest Neighbors falls to a mid-level position with 57.51% accuracy, while the linear-assumption Naive Bayes delivers by far the lowest performance with 37.40%. The Multi-Layer Perceptron (MLP), with an accuracy of 63.62%, indicates that neural networks cannot match the performance of ensemble methods at this data scale; meanwhile, AdaBoost offers a balanced alternative just below the top tier with 76.72%.

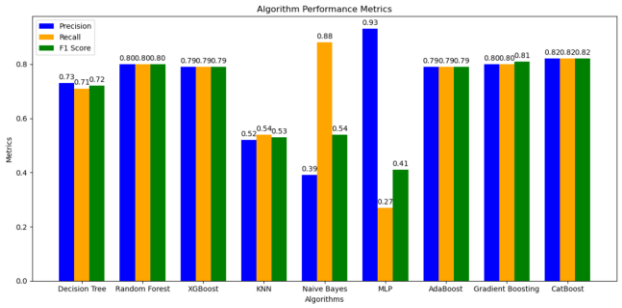

Fig. 4: Mod B(Algorithm performance metrics)
Source: created by the authors.

The algorithm performance bar chart for this mod (Figure 4) clearly reveals the strengths and weaknesses of the models across three metrics (precision, recall, and F_1-score).

3.1.3 Comparison of MOD A and MOD B

With 75% training, 25% test split, CatBoost achieved the highest F1-score at 0.82, establishing the most balanced accuracy error trade-off among all algorithms; Gradient Boosting followed very closely in second place with a score of 0.81.

Table 1. Mod A (%80 Train, %20 Test) Accuracy Table

ALGORITHMS	ACCURACY
Decision Tree	69.84%
Random Forest	78.31%
Multi-Layer Perceptron	63.62%
K-Nearest Neighbors	57.51%
Naive Bayes	37.40%
XGBoost	78.42%
AdaBoost	76.72%
Gradient Boosting	78.87%
CatBoost	**78.90%**

Source: created by the authors

Table 2. Mod B (%75 Train, %25 Test) Accuracy Table

ALGORITHMS	ACCURACY
Decision Tree	69.49%
Random Forest	78.31%
Multi-Layer Perceptron	73.11%
K-Nearest Neighbors	60.90%
Naive Bayes	37.40%
XGBoost	78.42%
AdaBoost	77.72%
Gradient Boosting	78.64%
CatBoost	**78.87%**

Source: created by the authors

When shifting to the 80% training,20% test scenario, the overall picture remained largely unchanged, though a slight shift at the top was observed: both Gradient Boosting and CatBoost fixed their F1-scores at 0.81, sharing the best performance; Random Forest and XGBoost steadily followed them in the 0.80–0.79 range (showing in Figure 5).

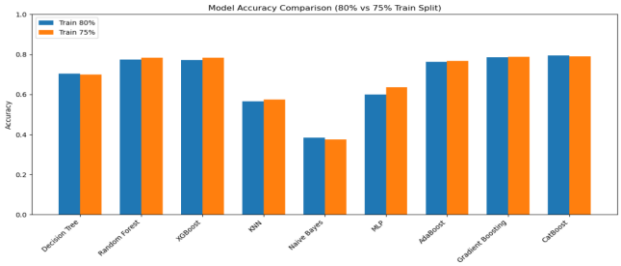

Fig. 5: Model accuracy comparison (80% vs 75% train split)
Source: created by the authors.

Table 3. Performance metric results for the 75% training set

x

	Precision	Recall	F1-Score
DecisionTree	0,73	0,71	0,72
Random Forest	0,80	0,80	0,80
XGBoost	0,79	0,79	0,79
KNN	0,52	0,54	0,53
Naive Bayes	0,39	0,88	0,79
MLP	0,93	0,27	0,41
AdaBoost	0,79	0,79	0,79
Gradient Boosting	0,80	0,80	0,81
CatBoost	0,82	0,82	0,82

Source: created by the authors

The increase in data volume made CatBoost's leadership less absolute while allowing Gradient Boosting to gain a slight edge; this indicates that tree-based ensemble methods scale similarly as both data quantity and class distribution change, and that their parameter sensitivities remain low.

Table 4. Performance metric results for the 80% training set

	Precision	Recall	F1-Score
DecisionTree	0,75	0,71	0,73
Random Forest	0,80	0,80	0,80
XGBoost	0,79	0,79	0,79
KNN	0,52	0,56	0,54
Naive Bayes	0,39	0,88	0,54
MLP	0,49	0,90	0,64
AdaBoost	0,79	0,79	0,79
Gradient Boosting	0,80	0,81	0,81
CatBoost	0,80	0,82	0,81

Source: created by the authors

4 Conclusion

The aim of this study is not only to explain some data mining algorithms in a detailed and understandable way but also to interpret their performance through an application. Additionally, the training datasets were split into two ratios, 80% and 75%, to obtain two different findings and compare them. As a sample dataset, the "Predict Students' Dropout and Academic Success" dataset from the UCI Machine Learning Repository was selected.

First, we introduced the algorithms used in our study and briefly discussed other important algorithms as needed. While interpreting these algorithms, we emphasized how important algorithm performance metrics are and what these metrics mean. Next, we split our dataset into two groups as 80% training and 20% testing. The algorithms used in our application were nine in total: Decision Tree, Random Forest, XGBoost, KNN, Naive Bayes, MLP, AdaBoost, Gradient Boosting, and CatBoost.

In conclusion, regarding Mod A, the model is extremely successful in distinguishing students in graduate status. A recall of 92% means that out of 100 students who graduated, 92 are correctly labeled as "graduate"; the model almost never misses examples belonging to this class. A precision of 81% means that one in five students predicted as "graduate" belongs to another class, indicating some "false graduate" labels. However, since recall is very high, if finding graduates is critical, the model is quite reliable. The 19% error in precision suggests that some students labeled as graduates may be dropouts or enrolled, implying that some features of the graduate class overlap with other classes and that additional discriminating variables (e.g., graduation date, percentage of credits completed) may be needed to reduce false positives.

For the dropout class, both precision and recall are around 82%; thus, the model misses one-fifth of the dropout students (in terms of recall) and one-fifth of those labeled as "dropout" are not dropouts (in terms of precision). This balanced picture shows that dropout class examples are sufficient in number and reasonably well separated in the feature space. However, an 18% error is still significant: since dropout students generally require early intervention as a high-risk group, the missed 18% (false negatives) can create gaps in early warning systems. Similarly, the 18% who are falsely labeled as dropouts may lead to unnecessary interventions (showing in Table 1 and Table 3).

Finally, commenting on Mod B, the model's overall accuracy is around 78.9%, showing strong performance, especially in the Graduate and Dropout classes. Regarding graduate students, the model correctly predicts 92% of those who graduated, almost completely capturing this group; also, four out of every five students labeled as "graduate" are truly graduates (82% precision). For dropout students, the table is balanced as well: recall is 82%, precision 80%, meaning the model both detects dropout status at a high rate and keeps false "dropout" labels at a reasonable level. These results make the system reliable in identifying graduates and monitoring dropouts, while errors could be further reduced by maintaining class balance and especially strengthening dropout-specific signals (showing in Table 2 and Table 4).

In future studies, other methods that were not addressed in this study can also be applied. Percentages that are not commonly used for training and test data can also be utilized.

References:

[1] Mahesh Mudhol Purushothama Gowda, (2004) Data Mining in the Process of Knowledge Discovery in Digital Libraries, 2nd Convention PLANNER, Manipur Uni., Imphal, page no 164-167, India.

[2] Siraj, F., & Abdoulha, M. A. (2011). Mining enrollment data using descriptive and predictive approaches. In Knowledge-Oriented Applications in Data Mining (pp. 53–72). InTech.

[3] P. Gundecha and H. Liu, Mining Social Media: A Brief Introduction. Tutorials in operations research, 2012.

[4] P. Ozer and I.G Sprinkhuizen-Kuyper, Data algorithms for classification, BSc Thesis Artificial Intelligence, Radboud University Nijmegen, January 2008.

[5] J. Han, M. Kamber and J. Pei. Data Mining Concepts and Techniques, Morgan Kaufmann, San Francisco, 2011.

[6] P. N. Tan, M. Steinbach and V, Kumar, Introduction to Data Mining, Pearson Addison Wesley, Boston, 2006.

[7] Ozdemir, O., Karakütük, F., & Kaya Karakütük, A. (2025). Veri Madenciliğinde Karar Ağaçları İndükleyicilerinin Likert Ölçekli Verilerde Sınıflandırma Performansı. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 29(1), 72-83, DOI: 10.19113/sdufenbed.1596624.

[8] Ferdi Karakutuk, Ozer Ozdemir, Asli Kaya Karakutuk, "Performance of Data Mining Algorithms for Classification of Likert Scale Data," WSEAS Transactions on Computers, vol. 24, pp. 127-135, 2025, DOI: 10.37394/23205.2025.24.12.

[9] M. J, Zaki and W. Meira Jr, Data Mining and Analysis-Fundamental Concepts and Algorithms, Cambridge University Press. New York, USA, 2014.

[10] E. Garcia, C. Romeo, S. Ventura and T. Calders, Drawbacks and solutions of applying association rule mining in learning management systems, Proceedings of the International Workshop on Applying Data Mining in e-Learning 2007, Crete Greece.

[11] S. M. Kamruzzaman, F. Haider and A. R. Hasan, Text Classification Using Data Mining, ICTM, 2005.

[12] Microsoft. Data Mining Algorithms. Analysis Services-Data Mining. 2016.

[13] Dekker, G. W., Pechenizkiy, M., & Vleeshouwers, J. M. (2009). Predicting students drop out: a case study. In T. Barnes, M. Desmarais, C. Romero, & S. Ventura (Eds.), Proceedings of the 2nd International Conference on Educational Data Mining, EDM 2009, July 1-3, 2009. Cordoba, Spain (pp. 41-50).

[14] *(text in Turkish)* Albayrak, .Y.S., & Yılmaz, Ö. K. (2009). Data Mining: Decision Tree Algorithms and an Application on ISE Data (Veri Madenciliği: Karar Ağaci Algoritmalari Ve Imkb Verileri Üzerine Bir Uygulama). Süleyman Demirel Üniversitesi İktisadi Ve İdari Bilimler Fakültesi Dergisi, 14(1), 31-52.

[15] Timarán Pereira R., Calderón Romero A. and Jiménez Toledo J. (2013). Extraction Student Dropout Patterns with Data Mining Techniques in Undergraduate Programs. In Proceedings of the International Conference on Knowledge Discovery and Information Retrieval and the International Conference on Knowledge Management and Information Sharing - Volume 1: KDIR, (IC3K 2013) ISBN 978-989-8565-75-4, pages 136-142, Vilamoura, Algarve, Portugal.

[16] Márquez-Vera, C., Cano, A., Romero, C., Noaman, A. Y. M., Mousa Fardoun, H., and Ventura, S. (2016) Early dropout prediction using data mining: a case study with high school students. Expert Systems, 33: 107–124, DOI: 10.1111/exsy.12135.

[17] Yan Li, Vineeth Rakesh, and Chandan K. Reddy. 2016. Project Success Prediction in Crowdfunding Environments. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (WSDM '16). Association for Computing Machinery, New York, NY, USA, 247–256, DOI: 10.1145/2835776.2835791.

[18] M. S. Oduro, H. Yu and H. Huang, "Predicting the Entrepreneurial Success of Crowdfunding Campaigns Using Model-Based Machine Learning Methods," in International Journal of Crowd Science, vol. 6, no. 1, pp. 7-16, April 2022, DOI: 10.26599/IJCS.2022.9100003.

[19] Yerlikaya-Özkurt, F.; Özbey, M.F.; Turhan, C. Modelling the mood state on thermal sensation with a data mining algorithm and testing the accuracy of mood state correction factor. New Ideas Psychol. 2025, 76, 101124, DOI: 10.1016/j.newideapsych.2024.101124.

[20] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 281–297.

[21] J.R. Quinlan, “Introduction of decision trees”, Machine Learning, vol.1, 1986, pp.81-106, DOI: 10.1007/BF00116251.

[22] Dunham, M. H. (2003). Data mining: Introductory and advanced topics. Prentice Hall, Pearson Education Inc., New Jersey.

[23] Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.

[24] Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. Wadsworth.

[25] Kass, G. V. (1980). An exploratory technique for investigating large quantities of categorical data. Applied Statistics, 29(2), 119–127, DOI: 10.2307/2986296.

[26] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[27] Rennie, J. D., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the poor assumptions of

naive bayes text classifiers. In Proceedings of the 20th International Conference on Machine Learning (ICML-03) (pp. 616-623), Washington DC, USA.

[28] Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 29(2), 103-130, DOI: 10.1023/A:1007413511361.

[29] Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.

[30] McCallum, A., & Nigam, K. (1998). A comparison of event models for naive bayes text classification. In AAAI-98 Workshop on Learning for Text Categorization (Vol. 752, pp. 41-48), Madison, Wisconsin, USA.

[31] Zhang, H.,Proceedings of 17th International Florida Artificial Intelligence Research Society Conference, Menlo Park, 12-14 May 2004, 562-567.

[32] Zhang, H. (2005). Exploring conditions for the optimality of naive Bayes. International Journal of Pattern Recognition and Artificial Intelligence, 19(02), 183-198, DOI: 10.1142/S0218001405003983.

[33] Rennie, J., & Srebro, N. (2005). Loss functions for preference levels: Regression with discrete ordered labels. In Proceedings of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling (pp. 180-186), Edinburgh, Scotland.

[34] Cover, T., & Hart, P. (1967). Nearest neighbour pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27, DOI: 10.1109/TIT.1967.1053964.

[35] Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 46(3), 175-185, DOI: 10.1080/00031305.1992.10475879.

[36] Kaya Karakutuk, A., & Ozdemir, O. (2025). A Novel Fuzzy Kernel Extreme Learning Machine Algorithm in Classification Problems. Applied Sciences, 15(8), 4506, DOI: 10.3390/app15084506.

[37] Kaya Karakutuk, A., Ozdemir, O., & Senturk, S. (2025). Optuna-Optimized Pythagorean Fuzzy Deep Neural Network: A Novel Framework for Uncertainty-Aware Image Classification. Applied Sciences, 15(20), 11097, DOI: 10.3390/app152011097.

[38] Minku, L., Cabral, G. G., Martins, M., & Wagner, M. (Eds.) (2023). Introduction to Computational Intelligence: An IEEE Computational Intelligence Society Open Book (1 ed.).

[39] Müller, A. C., & Guido, S. (2016). Introduction to machine learning with Python: A guide for data scientists. O'Reilly Media.

[40] Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems (2nd ed.). O'Reilly Media.

[41] Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences, 55(1), 119–139, DOI: 10.1006/jcss.1997.1504.

[42] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning (2nd ed.). Springer, DOI: 10.1007/978-0-387-84858-7.

[43] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794, New York, NY, USA, DOI: 10.1145/2939672.2939785.

[44] Zhang, Y., Chen, X., & Zhou, D. (2017). Regularization techniques for decision tree learning. arXiv preprint arXiv:1711.08905.

[45] Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. Frontiers in Neurorobotics, 7, 21, DOI: 10.3389/fnbot.2013.00021.

[46] Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., and Duchesnay É. 2011. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12, null (2/1/2011), 2825–2830.

[47] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: Unbiased boosting with categorical features. Advances in Neural Information Processing Systems, 31, 6638–6648.

[48] Hancock, J. T., & Khoshgoftaar, T. M. (2020). CatBoost for big data: An interdisciplinary review. Journal of Big Data, 7(1), 94, DOI: 10.1186/s40537-020-00369-8.

[49] Realinho, V., Vieira Martins, M., Machado, J., & Baptista, L. (2021). Predict Students' Dropout and Academic Success [Dataset]. UCI Machine Learning Repository, DOI: 10.24432/C5MC89.

[50] M.V.Martins, D. Tolledo, J. Machado, L. M.T. Baptista, V.Realinho. (2021) "Early prediction of student's performance in higher education: a case study" Trends and Applications in Information Systems and Technologies, vol.1, in Advances in Intelligent Systems and Computing series. Springer. DOI: 10.1007/978-3-030-72657-7_16.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed to the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.