International Journal of Emerging Engineering and Technology (IJEET)
ISSN (e): 2958-3764
Volume: 5, Number: 1-2, Pages: 73 - 81, Year: 2026 (Special Issue)

Benchmarking Seven Classifiers for Student Dropout Prediction on the UCI Higher Education Dataset

Bushra Raiz*, and Mudasser Naseer

Department of Computer Science, The University of Lahore, Lahore, 54000, Pakistan
*Corresponding author: Bushra Raiz (Email: bushra.riaz.dr.ds@gmail.com)

Received: 25/08/2026, Revised: 18/10/2026, Accepted: 21/10/2026

Abstract **—Attrition among students remains an ongoing problem in universities, so classifiers whose results can be compared and reproduced are needed. In this study, seven classical classifiers are tested using the public UCI Predict Students' Dropout and Academic Success data set containing 4,424 undergraduates with 36 attributes. The data set is recoded into 1,421 dropouts and 3,003 graduated or enrolled students. Logistic regression, decision tree, random forest, support vector machine, naive Bayes, k-nearest neighbors, and XGBoost are compared with one stratified 5-fold cross-validation scheme. Class weights are applied to the five classifiers that support them while the decision threshold for each classifier is optimized for F1 score. XGBoost yields the highest results (accuracy 0.8804, ROC-AUC 0.9242, F1-score 0.8076 at a decision threshold of 0.53), followed closely by random forest and logistic regression. A repeat with fixed seeds and in-fold scaling gives accuracy, ROC-AUC, and F1-score close to those reported (within 0.0012), and the bootstrap 95% confidence intervals for the top three classifiers overlap, so these models are not reliably separated. Nested cross-validation caused a drop in the XGBoost F1-score of only 0.0002 on the primary split and of 0.0037 on average over six splits. Gain, permutation, and SHAP importance all show that approved second-semester units and tuition fee status play the most important role, but differ on the lower-ranked features. Without the second-semester features, the XGBoost ROC-AUC dropped from 0.9241 to 0.9037. These results refer to one particular dataset of one institution.**

Index Terms**—Class imbalance, Decision threshold optimization, Educational data mining, Student dropout prediction, XGBoost**

I. INTRODUCTION

STUDENT dropout leads to decreased numbers of graduates, wastage of resources by the institution, and is costly for the students who drop out. Many institutions have interest in early warning systems that would be able to identify students at risk, who could still benefit from additional assistance. Machine learning (ML) classifiers based on administrative and academic data are commonly used for building such tools [1]-[4]. There is great variation in the literature, though, in terms of the data, outcome definitions, methodology, and metrics used, which makes comparison of the reported scores difficult [5]. Most of the models are developed and validated on one institution's data [6].

Public data benchmarks alleviate this issue. The dataset "Predict Students' Dropout and Academic Success" made available by Realinho et al. via the UCI Machine Learning Repository includes information about 4,424 undergraduate students of a Portuguese polytechnic institute with various demographic, socioeconomic, macroeconomic, and academic features [7], [8]. Since the data are public, the results for this dataset are reproducible and comparable, and the dataset has been utilized in several recent papers [9]-[12]. These papers employ different targets (three classes or only dropout/graduate), different splits, and resampling approaches. In any case, a clearly documented baseline on standard classifiers is still useful.

This work provides such a baseline. It is taken from the MS thesis of the first author where this dataset was used as an external benchmark. Another manuscript investigates a local survey dataset of students from the same thesis; both papers use different datasets and have different results, tables, and figures, and only the publicly available UCI dataset (id 697, doi 10.24432/C5MC89) is used here. In this manuscript, the three-class classification problem is converted to the dropout vs. non-dropout binary classification problem. Seven classical classifiers are analyzed based on the same cross-validation methodology, and the threshold is optimized for every classifier based on the maximum F1 score. The best-performing model is interpreted using the gain, permutation, and SHAP importance. Thus, the goal of the work is a clear point of reference and not a novel approach. The contributions are as follows.

- The comparative performance of logistic regression (LR), decision tree (DT), random forest (RF), support vector machine (SVM), naive Bayes (NB), k-nearest neighbors (KNN), and extreme gradient boosting (XGBoost) classifiers on the binary problem from the UCI dataset according to the same cross-validation method, class weighting, and performance metrics.
- The analysis of the optimal operating point, including its confusion matrix, ROC curve, and behavior of precision, recall, and F1 score depending on the decision threshold.



This work is licensed under a Creative Commons AttributionShareAlike4.0 International License, which permits Unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited

- The explanation of features important for the XGBoost classifier, comparing the results with those provided by the authors of the dataset and subsequent research papers.
- An explanation of the limitations of the protocol, notably that the threshold values have been fine-tuned using the same out-of-fold predictions as those used for evaluation.
- Robustness checks of the protocol: bootstrap confidence intervals, a nested test of the threshold, permutation and SHAP importance, and a modification where second-semester features were omitted.

Section II provides the literature review. Section III provides details of the data used, models, validation protocol, and metrics used. Section IV describes the results obtained. Section V provides a comparison with previous studies and limitations, while Section VI provides conclusions.

II. Related Work

A. Machine Learning for Dropout Prediction

The early dropout prediction models largely depended on logistic regression and other linear techniques, which were easily interpretable but could not identify interactions between predictors. Kemper et al. applied logistic regression and decision tree algorithms to examination data collected at the Karlsruhe Institute of Technology and found that the accuracy was higher than 83% after the first semester and up to 95% after three semesters [1]. Sandoval-Palis et al. compared logistic regression with a neural network by applying them to the data of 2,097 university leveling students [2]. Kabathova and Drlik compared different classifiers on course-level data from four academic years and found accuracy from 77% to 93% in the following year [3]. Fernández-García et al. created models for different stages of the degree using solely academic variables [4]. Ensemble learning approaches have been broadly applied now. Niyogisubizo et al. combined the outputs of random forest and XGBoost in a neural network [13], and Kim et al. introduced a hybrid approach with an emphasis on high precision and recall in predicting dropouts [14].

There are two main challenges that appear in this line of research frequently. First, the class imbalance problem, since usually dropouts form the minority class. Both data-level solutions such as the synthetic minority over-sampling technique (SMOTE) and algorithmic level solutions like class weights have been employed [15], [16], as well as a comparative study of balancing techniques for dropout data by Mduma [17]. Second, there is the issue of interpretability. The institution needs to understand why a certain student has been flagged, which resulted in explainable models and decision support systems for dropouts [18], [19]. A systematic review of studies from 2012 to 2024 identified decision trees, logistic regression, and random forests as the most popular models and SHAP and LIME as the most applied explanation methods [5].

B. Studies on the UCI Dataset

This dataset was introduced by Martins et al., who conducted research on early prediction of performance of students at the Polytechnic Institute of Portalegre [9], and later described in a data descriptor by Realinho et al. [7]. In the descriptor, the problem is formulated as a three-class classification problem (dropout, enrolled, graduate), mentioning class imbalance and high correlation of features in the first and second semesters, along with reporting permutation importance values for RF and three boosting methods [7]. Villar and de Andrade retained the three classes, used the SMOTE technique, and tuned DT, SVM, RF, and four boosting methods using Optuna, where LightGBM and CatBoost showed superior performance [12]. On the other hand, Romero and Liao modeled the problem of dropout and graduate classification and hence did not consider enrolled students in their task; they used an 80/20 split ratio, where the XGBoost model showed the highest F1-score (0.904) and the RF model had the highest AUC score (0.935) [10]. Ridwan et al. trained XGBoost under several train/test ratios and obtained 88% precision and an 81% F1-score with the 80:20 split [11].

The results are not directly comparable with one another because they are different not only with respect to the target (three classes, dropout vs graduate or dropout vs all others) but also with respect to the validation scheme (single split or cross-validation) and resampling or tuning procedure. This paper fixes one target, that is dropout vs non-dropout with students who are still enrolled being classified as non-dropouts, and considers all seven models according to the same protocol.

C. Threshold Selection

Most classifiers provide a score/probability as the output and the cutoff value transforms the score/probability into the final decision. The standard cutoff of 0.5 is rarely optimal for imbalanced tasks or for metrics such as F1. Lipton et al. analyzed the optimal cutoff maximizing F1 and demonstrated that the choice of the cutoff value depends on the distribution of the scores and the best achievable F1 [20]. The choice of the cutoff based on the data used for reporting of the classifier's performance is a form of model selection and may lead to overestimation [21], [22]. Section V returns to this point.

III. Materials and Methods

A. Dataset and Target Definition

The dataset used for this study is the Predict Students' Dropout and Academic Success dataset from the UCI Machine Learning Repository [8]. Each of the 4,424 observations represents one student enrolled in one of 17 undergraduate degrees during the period between 2008/2009 and 2018/2019 [7]. There are 37 columns in the file: 36 features and the target variable. These features include demographic information (such as the age of the students at enrollment, their gender and nationality), socioeconomic information (including parents' qualification level and occupation, whether or not the students were debtors, tuition fee status and whether or not they had scholarships), macroeconomic information (unemployment, inflation rates and gross domestic product), academic information at enrollment and the number of enrolled, evaluated, credited and approved curricular units and grades in the first and second semesters [7], [8]. There are no missing values in the dataset [7]. The present paper relies on this single

data source, which is a publicly available file; the survey data of the thesis are used in a separate companion paper and are not considered in the current analysis.

The original target variable has three classes: Graduate (2,209 students), Dropout (1,421) and Enrolled (794). As the goal is to find potential dropouts, the target was transformed into Dropout = 1 and Non-dropout (Graduated/Enrolled) = 0. The positive class thus contains 1,421 examples (32.1%) and the negative class 3,003 (67.9%), as shown in Table I. A model that would always predict non-dropout achieves an accuracy of 0.6788, the baseline against which all other accuracies mentioned are compared.

TABLE I
Class Distribution of the UCI Dataset and the Binary Target

Original class	Students	Share (%)	Binary label
Graduate	2,209	49.9	Non-dropout (0)
Enrolled	794	17.9	Non-dropout (0)
Dropout	1,421	32.1	Dropout (1)
Total	4,424	100.0	3,003 (0) / 1,421 (1)

Counts from the thesis and the data descriptor [7]; shares computed from the counts.

B. Preprocessing and Classifiers

The pipeline is identical to the one from the thesis' main model and was not changed in any way when applied to the UCI dataset. In the re-run, the features were standardized for LR, SVM, KNN, and NB, while the tree-based classifiers were provided with the raw values. Seven models were fit: LR, a basic probabilistic baseline; DT, a single rule-based tree; RF, a bagged ensemble of trees [23]; SVM with probabilities enabled; NB; KNN; and XGBoost, a regularized gradient boosting algorithm [24]. Class weighting was performed for LR, DT, RF, and SVM (class_weight = "balanced" in scikit-learn [25]), and for XGBoost via the scale_pos_weight parameter, being equal to the ratio of negative to positive students in each training fold [24]. No class weighting was done for KNN and NB as it is not an available option for them; thus, in these two models, tuning of the decision threshold was the only method to handle the class imbalance. The categorical features were employed using their integer codes from the UCI file and not one-hot encoded, and no imputation, resampling, or feature selection was applied.

The settings of the re-run are shown in Table II below. There is a lack of information about the hyperparameters, random seeds, and software versions used in the thesis, thus these settings were recovered for the re-run. The settings of the XGBoost model were selected among the grid of 96 options (the number of trees, learning rate, depth, and row sampling) by picking the configuration whose pooled out-of-fold results were closest to those in the thesis table. For the RF model the forest size and for KNN the number of neighbors were selected in the same way from small grids. Thus, the goal was to reproduce the thesis table rather than maximize performance, and no other tuning was performed. The random seed 42 was used for the data split and every model. All code was run using Python 3.13.5 and the following libraries: scikit-learn 1.9.1, XGBoost 3.4.1, SHAP 0.54.1, NumPy 2.5.4, pandas 3.0.6, and SciPy 1.18.1.

TABLE II
Settings of the Seven Classifiers in the Re-run

Model	Class weights	Scaling	Main settings
LR	balanced	in-fold	L2 penalty, C = 1, lbfgs solver, max_iter = 1000
DT	balanced	none	Gini criterion, fully grown (no depth limit)
RF	balanced	none	500 trees, Gini criterion, 6 features per split
SVM	balanced	in-fold	RBF kernel, C = 1, gamma = "scale", Platt-scaled probabilities
KNN	not supported	in-fold	k = 7, Euclidean distance, uniform weights
NB	not supported	in-fold	Gaussian, var_smoothing = 1e-9
XGBoost	scale_pos_weigh t	none	500 trees, learning rate 0.03, max depth 5, no row or column subsampling

In-fold: standard scaler fitted on the training part of each fold. Other parameters are library defaults.

C. Validation Protocol and Threshold Tuning

All the models were validated using the same stratified 5-fold cross-validation procedure (shuffle, seed 42), where the class proportions in each fold are equal to those in the whole dataset. Each student had only one out-of-fold predicted probability of dropping out, $\hat{p}$, from the model that did not have this student in its training process. The confusion matrix of the best model consists of all 4,424 students (Section IV); therefore, all metrics were calculated from this pooled set of out-of-fold predictions. Subsequently, for each of those models, the decision threshold τ was varied from 0.05 to 0.95 by increments of 0.01, the F1-score was computed for each threshold, and the one maximizing the F1-score was chosen; ties were broken by preferring the lower threshold. A student is considered to be a dropout if the predicted probability reaches the threshold τ, as defined by (1). The optimal value of the threshold is given by (2).

$$\hat{y} = \begin{cases} 1, \text{if } \hat{p} \geq \tau \\ 0, \text{otherwise} \end{cases} \quad (1)$$

$$\tau^* = \underset{\tau \in [0.05,\, 0.95]}{\arg\max} \; \mathrm{F1}(\tau) \quad (2)$$

Here, $\mathrm{F1}(\tau)$ denotes the F1-score of the out-of-fold predictions with the decision threshold equal to τ. Accuracy, precision, recall, and F1 are provided at τ^*, while ROC-AUC is independent of the threshold. In the re-run, each preprocessing step was included into a scikit-learn pipeline [25] that was trained using only the training part of each fold: the standard scaler learned the means and standard deviations of the training students and applied them to the held-out students. Neither imputation, nor resampling, nor feature selection were applied, so nothing derived from the held-out students was included in the training process except the threshold, whose choice based on the pooled predictions is studied in Section IV-D. As a check, standardizing all students prior to cross-validation changed the ROC-AUC of KNN from 0.8595 to 0.8608 and the F1-score of SVM from 0.7899 to 0.7894. The in-fold version yielded the same values as the thesis did, which is consistent with in-fold scaling in the original experiments, although it does not prove it.

D. Evaluation Metrics

Where dropout is the positive class, TP is the number of dropouts classified as dropouts, FN is the number of dropouts misclassified as non-dropouts, FP is the number of non-dropout students wrongly classified as dropouts, and TN is the number of non-dropout students classified as non-dropouts. All metrics follow their standard definitions [26], [27]. Accuracy (3) is the proportion of correct classifications. Precision (4) is the proportion of flagged students who are actual dropouts. Recall (5) is the proportion of dropouts flagged, and specificity (6) is the proportion of non-dropout students correctly not flagged. The F1-score (7) is the harmonic mean of precision and recall.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (3)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (6)$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

The ROC curve plots the True Positive Rate (TPR), which equals recall, as a function of the False Positive Rate (FPR) at all possible thresholds, and ROC-AUC is the area under it [27], as shown in (8) and (9).

$$\text{ROC-AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d\text{FPR} \quad (8)$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \ \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (9)$$

An ROC-AUC of 0.5 means random ordering and 1.0 implies perfect separation. However, on imbalanced data, accuracy can have high values even though only a few cases of the minority class are detected [16], [28]. Therefore, recall, precision, and F1 are also reported.

E. Explainability

As for the best-performing model (XGBoost), the gain is used to measure global feature importance. The gain reflects the average reduction in training loss gained from the splits using the feature [24]. In other words, the gain measures feature importance from the point of view of the fitted trees. The direction of the effect is not shown, and the features with numerous unique values might have an advantage [29]. Therefore, it is applied to describe the model, not to make causal claims. Since gain depends on the way that the trees have been constructed, two additional measures were calculated in the re-run of the analysis. Permutation importance [30] is the average decrease in the ROC-AUC when the values of one feature have been shuffled in a hold-out fold, and it was calculated with 10 repeats for each of the five fold models and averaged. SHAP importance [31], [32] is the average of absolute SHAP values of a feature among the hold-out students using the tree explainer with the fold models. Gain was calculated twice: once for the model fitted to all 4,424 students, and once by averaging over the five fold models, because the thesis does not specify which one of the two provided the figure for importance.

F. Uncertainty and Robustness Checks

Uncertainty due to sampling was evaluated by the percentile bootstrap [33]. The 4,424 students were resampled 1,000 times with replacement, and each metric was recalculated for each resample using the pooled out-of-fold predictions of the model at a fixed threshold. The 2.5th and 97.5th percentiles define the 95% confidence interval, and differences between two models were calculated for the same resamples. These confidence intervals reflect only the variance of the evaluation sample and do not contain the variance associated with retraining of models.

Bias caused by the optimal threshold choice on the evaluation predictions was studied with nested cross-validation [21], [22]. In each of five outer folds, the inner stratified 5-fold cross-validation on outer training students provided out-of-fold predictions, where the F1-optimal threshold was determined. Then the model retrained on the whole outer training dataset was thresholded by this value on the outer test fold. Outer pooled predictions provide an F1-score without the threshold tuning based on the test students. The main split was used for all seven models, but five more outer splits (seeds 43 to 47) were additionally performed for XGBoost.

Finally, sensitivity to late data was estimated for two reduced feature sets, using the full protocol with threshold optimization. The first feature set discards the six second-semester attributes and has 30 features, which are known at the end of the first semester. The second one discards all twelve curricular-unit attributes of both semesters and has 24 features, which are known at enrollment.

IV. RESULTS

A. Comparison of the Seven Classifiers

The results for the accuracy, ROC-AUC, F1-score, and F1-optimal threshold are presented in Table III, sorted by accuracy. The highest score according to these three criteria belongs to the XGBoost model (0.8804 accuracy, 0.9242 ROC-AUC, and 0.8076 F1-score at $\tau^* = 0.53$). The RF model followed closely (0.8784, 0.9227, 0.8032), and the third place belongs to the LR model (0.8732, 0.9184, and 0.8000). There is little difference between the XGBoost and RF models: 0.0020 of accuracy, 0.0015 of ROC-AUC and 0.0043 of F1-score. Even the linear model LR differs from XGBoost in ROC-AUC by no more than 0.0058. SVM took the fourth place, with ROC-AUC being equal to 0.9144 and F1-score to 0.7899.

TABLE III
Cross-Validated Results of the Seven Classifiers on the Binary UCI Task

Model	Accuracy	ROC-AUC	F1-score	τ^*
XGBoost	0.8804	0.9242	0.8076	0.53
RF	0.8784	0.9227	0.8032	0.46
LR	0.8732	0.9184	0.8000	0.56
SVM	0.8651	0.9144	0.7899	0.39
KNN	0.8291	0.8595	0.7206	0.29
NB	0.8192	0.8585	0.7286	0.17
DT	0.8047	0.7738	0.6934	0.05

τ^*: F1-optimal decision threshold. Accuracy and F1-score are computed at τ^*; ROC-AUC is threshold-free. Thesis values, rounded to four decimals.

The weaker models can be distinguished into one group. The ROC-AUC scores of KNN and NB were 0.8595 and 0.8585 and the F1-scores 0.7206 and 0.7286, respectively, so the accuracy was lower in NB but the F1-score was higher. DT turned out to be clearly the weakest model with ROC-AUC of 0.7738 and F1-score of 0.6934. The selected DT threshold is 0.05, which corresponds to the minimum threshold value of the search grid, but this does not imply that a smaller threshold can improve the results. A fully grown tree predicts leaf probabilities of exactly 0 or 1, and thus any threshold between 0 and 1 produces the same predictions and the same F1-score. The first grid value is used as default, and thus the threshold of DT is uninformative. The low threshold of NB (0.17) is likewise the reflection of poorly calibrated probabilities. All the seven models have an accuracy greater than the majority-class accuracy of 0.6788, but the margin for DT is modest.

The numbers from Table III were obtained during the experiments performed in the original thesis. Since there was no storage of hyperparameters, seeds, and predictions in the original thesis, all subsequent analyses were conducted using the re-run described in Section III-B. The re-run replicated Table III up to four decimal places for LR, SVM, KNN, and NB, including the thresholds, and had minor differences for the other three models. For XGBoost it resulted in 0.8802, 0.9241, and 0.8077 at the same threshold of 0.53, and its confusion counts (1,113 true positives and 222 false positives) are 3 and 4 students away from the numbers in Table IV (1,110 and 218). For RF it resulted in 0.8775, 0.9226, and 0.8042 at a threshold of 0.50 instead of 0.46, and for DT 0.8056, 0.7747, and 0.6946. The largest gap between a thesis value and its re-run value in any accuracy, ROC-AUC, or F1-score is 0.0012. Table III and the findings based on it (Table IV and Figs. 1 to 4) remain unchanged from the thesis, and Tables V to VIII are based on the re-run.

B. Operating Point of the XGBoost Model

In Fig. 1, the confusion matrix of XGBoost for $\tau^* = 0.53$ is presented, and Table IV gives the counts and the metrics based on them. Among 1,421 dropouts, 1,110 cases were detected while 311 cases were missed; hence, the recall equals 0.7811. Out of 1,328 students flagged as dropouts, 218 were non-dropouts, and therefore the precision is 0.8358. In turn, among 3,003 non-dropouts, 2,785 were correctly left unflagged, and thus the specificity equals 0.9274. Overall, 3,895 students out of 4,424 were classified correctly, and this corresponds to the accuracy of 0.8804 reported in Table III; additionally, the counts confirm the F1-score of 0.8076. On the whole, 30.0% of all students were flagged as dropouts, slightly fewer than the actual dropout rate of 32.1%.

The ROC curve of XGBoost is shown in Fig. 2; it demonstrates a steep increase at low false positive rates, which means that the students with the highest predicted risk are mostly true dropouts. Fig. 3 depicts how the precision, recall, and F1-score depend on the threshold. Recall falls and precision rises as τ increases; the curves intersect at about 0.46, where they both have a value of roughly 0.80.

TARGET/ OUTPUT	Class0	Class1	SUM
Class0	2785 62.95%	311 7.03%	3096 89.95% 10.05%
Class1	218 4.93%	1110 25.09%	1328 83.58% 16.42%
SUM	3003 92.74% 7.26%	1421 78.11% 21.89%	3895 / 4424 88.04% 11.96%

Fig. 1. Confusion matrix of XGBoost at $\tau^* = 0.53$ over the pooled out-of-fold predictions for all 4,424 students. Columns are the true classes and rows the predicted classes (Class0: non-dropout; Class1: dropout). Percentages in the margins give per-row and per-column correct (upper) and incorrect (lower) shares.

TABLE IV
Confusion Counts and Derived Metrics of XGBoost at $\tau^* = 0.53$

Quantity	Value	Quantity	Value
TP	1,110	Accuracy	0.8804
FN	311	Precision	0.8358
FP	218	Recall	0.7811
TN	2,785	Specificity	0.9274
Flagged	1,328	F1-score	0.8076

Counts from Fig. 1 (pooled out-of-fold predictions, n = 4,424); metrics computed from the counts with (3) to (7).

The F1 curve flattens out near its maximum: from Fig. 3 we can see that its values stay close to 0.80 for τ values ranging from about 0.4 to about 0.7. Thus, the choice of 0.53 is not critical for F1, but it is important for the trade-off between missed dropouts and false alarms.

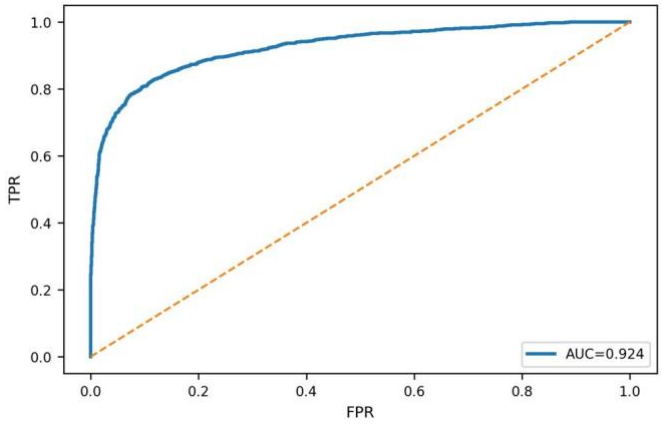


Fig. 2. ROC curve of XGBoost for the dropout class (TPR: true positive rate; FPR: false positive rate). The dashed line is a random classifier. AUC = 0.924.

C. Bootstrap Confidence Intervals

Table V presents the point estimates and 95% bootstrap intervals of the re-run of all seven models. The bootstrap intervals of XGBoost, RF, and LR overlap widely for every metric; for instance, F1-score of XGBoost is 0.8077 (0.7911 to 0.8237), and F1-score of LR is 0.8000 (0.7825 to 0.8166). Pairwise comparisons on the same set of resampled data give the same result: the accuracy interval of XGBoost minus RF (-0.0036 to 0.0088), the F1-score interval (-0.0066 to 0.0131), and the ROC-AUC interval (-0.0027 to 0.0059) all contain zero.

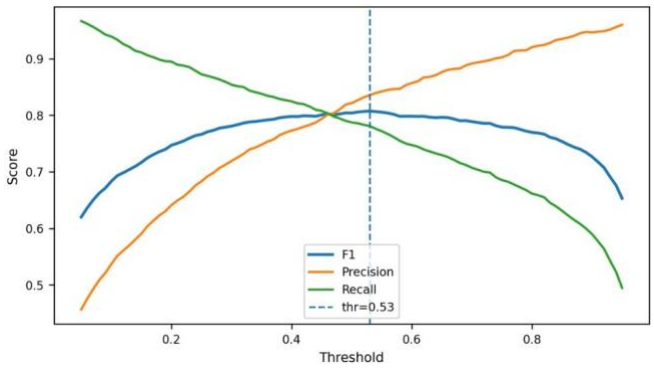


Fig. 3. Precision, recall, and F1-score of XGBoost versus the decision threshold. The dashed vertical line marks the F1-optimal threshold of 0.53.

In addition, the accuracy interval of XGBoost minus LR (-0.0009 to 0.0142) and the F1-score interval (-0.0048 to 0.0191) contain zero, while the ROC-AUC interval (-0.00004 to 0.0113) contains zero only barely. Therefore, the three best-performing models cannot be reliably separated from each other for this particular dataset. For XGBoost, the other main metrics include the precision 0.8337 (0.8140 to 0.8540), recall 0.7833 (0.7624 to 0.8037), and specificity 0.9261 (0.9169 to 0.9358). The confidence interval for the resampled confusion matrix from the thesis run (see Table IV) is 0.8143 to 0.8553 for the precision 0.8358 and 0.7602 to 0.8050 for the recall 0.7811.

TABLE V
Bootstrap 95% Confidence Intervals of the Re-run

Model	Accuracy	F1-score	ROC-AUC
XGBoost	0.8802 [0.8702, 0.8897]	0.8077 [0.7911, 0.8237]	0.9241 [0.9141, 0.9331]
RF	0.8775 [0.8678, 0.8868]	0.8042 [0.7889, 0.8200]	0.9226 [0.9125, 0.9318]
LR	0.8732 [0.8626, 0.8836]	0.8000 [0.7825, 0.8166]	0.9184 [0.9082, 0.9278]
SVM	0.8651 [0.8549, 0.8752]	0.7899 [0.7730, 0.8075]	0.9144 [0.9049, 0.9238]
KNN	0.8291 [0.8180, 0.8400]	0.7206 [0.7006, 0.7410]	0.8595 [0.8461, 0.8724]
NB	0.8192 [0.8065, 0.8293]	0.7286 [0.7090, 0.7468]	0.8585 [0.8456, 0.8708]
DT	0.8056 [0.7945, 0.8171]	0.6946 [0.6754, 0.7145]	0.7747 [0.7617, 0.7876]

Re-run, pooled out-of-fold predictions at the fixed F1-optimal threshold of each model. Percentile intervals from 1,000 bootstrap resamples of the 4,424 students.

D. Nested Threshold Selection

Table VI compares the F1-score at the threshold chosen on the same pooled predictions with the nested F1-score. In case of XGBoost the nested F1-score is 0.8075 against 0.8077, the difference being -0.0002; on average, over the six outer splits, the F1-score decreases from 0.8021 to 0.7984 (-0.0037, with a range of -0.0002 to -0.0065). The decreases for RF, LR, and SVM are between 0.0010 and 0.0015. KNN and DT models do not show any changes since their respective thresholds remain the same across all inner splits (0.29 for KNN, and 0.05 for DT, where the threshold does not affect the results). NB model demonstrates the greatest reduction in F1-score (-0.0087), which agrees with its highly variable inner thresholds (0.05 to 0.33) and its probably poorly calibrated probabilities. ROC-AUC does not depend on the threshold and remains unaffected. Thus, the bias resulting from threshold selection is small; the order of the four leading models is the same; only NB and KNN models swap places.

TABLE VI
Nested versus Same-Prediction Threshold Selection

Model	F1, same predictions	F1, nested	Change	Inner thresholds
XGBoost	0.8077	0.8075	-0.0002	0.47 to 0.53
RF	0.8042	0.8027	-0.0015	0.45 to 0.56
LR	0.8000	0.7990	-0.0010	0.55 to 0.58
SVM	0.7899	0.7884	-0.0014	0.38 to 0.40
KNN	0.7206	0.7206	0.0000	0.29
NB	0.7286	0.7199	-0.0087	0.05 to 0.33
DT	0.6946	0.6946	0.0000	0.05
XGBoost, 6 splits	0.8021	0.7984	-0.0037	0.45 to 0.64

F1-scores of the re-run. The last row is the mean over the main split and five further outer splits (seeds 43 to 47). Inner thresholds are the values selected in the five inner runs of the main split (all inner runs for the last row).

E. Feature Importance

Fig. 4 shows the top ten features based on gain in the XGBoost classifier. The number of approved curricular units in the second semester has a far larger gain than any other feature, followed by tuition fee status (tuition fees up to date). Next are the number of approved and enrolled units in the first semester and the number of enrolled units in the second semester. Debtor status, scholarship status, number of credited units in the second semester, gender, and age at enrollment complete the list. Overall, thus, the model is mostly based on academic achievements of the first year and financial indicators; none of the ten highest gains belongs to a macroeconomic factor, although the unemployment rate ranks seventh by SHAP importance (Table VII).

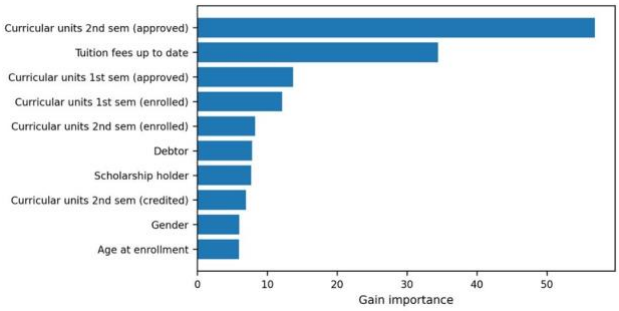


Fig. 4. Ten most important features of the XGBoost model, measured by gain (thesis run). Feature names follow the UCI variable names.

The thesis does not clarify whether the gain in Fig. 4 has been obtained using the model fitted on the entire data set or the fold models. Re-running results in nearly identical ordering (Spearman correlation 0.953, with nine out of ten leading features in common) for the two approaches, and each shares nine out of the ten features of Fig. 4; however, only age at enrollment (which is twelfth in both cases) is missing. Table VII shows that approved second-semester units and tuition fee status occupy the first two positions in all measures; moreover, the permutation importance of the first of them (decrease in ROC-AUC of 0.1568) is much higher than that of any other

feature. Underneath these two features the measures diverge, with rank correlation over all 36 features being equal to 0.78 for gain and permutation importance, 0.57 for gain and SHAP, and 0.80 for permutation and SHAP importance, and the top-ten lists sharing six, four, and seven features, respectively.

TABLE VII
Importance Measures for Selected Features

Feature	Gain rank (share)	Permutation rank (AUC drop)	SHAP rank (mean abs.)
Units 2nd sem. (approved)	1 (25.3%)	1 (0.1568)	1 (1.416)
Tuition fees up to date	2 (12.7%)	2 (0.0336)	2 (0.489)
Units 1st sem. (approved)	5 (3.2%)	3 (0.0107)	3 (0.355)
Course	13 (2.0%)	4 (0.0086)	4 (0.283)
Age at enrollment	12 (2.1%)	7 (0.0044)	5 (0.272)
Unemployment rate	27 (1.3%)	18 (0.0019)	7 (0.190)
Units 1st sem. (enrolled)	3 (6.8%)	8 (0.0041)	16 (0.122)
Units 2nd sem. (credited)	4 (3.5%)	17 (0.0019)	25 (0.067)

Re-run XGBoost; ranks among 36 features. Gain is from the model fitted to all students; permutation and SHAP importance are averages over the five fold models on held-out students. SHAP values are in log-odds units.

F. Early-Warning Variants

Table VIII presents the impact that is incurred through the omission of late features. Through excluding the six second-semester features, XGBoost manages to attain a ROC-AUC of 0.9037 (bootstrap 95% confidence interval: 0.8926 to 0.9144) and F1-score of 0.7722 (0.7549 to 0.7910) at a threshold of 0.51, where it is 0.0204 and 0.0355 below the model with all attributes included, respectively. For the case when only the enrollment-time features are used, the ROC-AUC of XGBoost is 0.8182 (0.8048 to 0.8314) and the F1-score is 0.6497 (0.6314 to 0.6694). With these enrollment-time features XGBoost detects 74.5% of the students who drop out, but only 57.6% of those flagged are actual dropouts. Hence most of the ranking capacity is retained upon omission of the second-semester features, while enrollment-time information alone is clearly less informative.

TABLE VIII
Early-Warning Variants without Late Attributes

Feature set	Model	ROC-AUC	F1-score	Recall	Precision
All 36 features	XGBoost	0.9241	0.8077	0.7833	0.8337
	RF	0.9226	0.8042	0.7833	0.8263
	LR	0.9184	0.8000	0.7896	0.8107
No 2nd semester (30)	XGBoost	0.9037	0.7722	0.7635	0.7811
	RF	0.9011	0.7677	0.7558	0.7800
	LR	0.8978	0.7657	0.7544	0.7774
Enrollment only (24)	XGBoost	0.8182	0.6497	0.7445	0.5763
	RF	0.8128	0.6457	0.6707	0.6225
	LR	0.7928	0.6203	0.6270	0.6136

Re-run with the full protocol, including threshold tuning, for each feature set. Recall and precision are at the F1-optimal threshold of each model.

V. Discussion

A. Comparison with Prior Work

Table IX presents the current findings side-by-side with those obtained from the papers discussed in Section II. The most comparable study is that of Romero and Liao, who obtained an F1-score of 0.904 for XGBoost and an AUC of 0.935 for RF [10]. Their problem considered no enrolled students; only graduates and dropouts were involved. The current problem includes 794 enrolled students in the negative class. The enrolled students did not graduate yet, and some may show similar records in the first year as the dropouts. Thus, the binary classification used in this case is likely more difficult. Together with the different validation design, this difference in target is a plausible reason for the lower F1-score of 0.8076, although the two studies cannot be compared formally. Our precision of 0.8358 is 0.0442 below the 88% attained by Ridwan et al. using XGBoost on the same dataset [11], and its 95% bootstrap interval (0.8143 to 0.8553), calculated from our predictions, does not cover 0.88, while our F1-score of 0.8076 is only 0.0024 below their 81%. These differences are, however, not to be over-interpreted, because their values are rounded to whole percentages and come from another splitting approach.

TABLE IX
Studies Using the UCI Dropout Dataset

Study	Target and design	Reported result
Realinho et al. [7]	Data descriptor; three classes	Permutation importance; no benchmark score
Villar and de Andrade [12]	Three classes; SMOTE; Optuna tuning	LightGBM and CatBoost best (per-class F1)
Romero and Liao [10]	Dropout vs. graduate; 80/20 split	XGBoost F1 0.904; RF AUC 0.935
Ridwan et al. [11]	XGBoost; several train/test ratios	80:20 split: precision 88%, F1 81%
This study	Dropout vs. non-dropout; stratified CV; F1-tuned τ	XGBoost F1 0.8076; AUC 0.9242

Results are as reported by each study and are not directly comparable because targets and designs differ.

The order of the classifiers coincides with previous findings. The boosting-based algorithms have been the most effective in this dataset for three-class [12] and dropout vs. graduate settings [10]; the same pattern is observed here. The close values of XGBoost, RF, and LR are also informative in this regard. On these tabular data, a properly weighted linear model is nearly as discriminative as a boosted ensemble; therefore, the usage of LR as a transparent baseline is justified. Bootstrap intervals in Table V provide additional support for this interpretation: the paired differences between these three models include zero in the case of accuracy and F1 score; for ROC-AUC, the interval for XGBoost vs. LR barely includes it (-0.00004 to 0.0113).

As regards feature importance, it is also in line with previous results. In the data descriptor, the approved curricular units in both semesters and tuition fee status were included in the list of the five features that have been important for all four algorithms [7]. For Romero and Liao, second semester results, credited units, tuition fee status, debtor status, scholarship status, and

age at enrollment were important for XGBoost [10]. All of these variables are found among the top 10 in Fig. 4. Consistency of findings between the studies employing different algorithms and importance measures suggests that these features carry real signal in this particular data set, but it does not prove that the ranking is independent of the methodology. In the re-run, gain, permutation and SHAP importance methods agree on the top-two features while diverging after them: credited second-semester units come fourth in terms of gain importance and 17th in terms of permutation importance; the unemployment rate appears seventh in SHAP importance ranking and 27th in gain importance ranking (Table VII). The lower-ranked features thus should be interpreted in the context of one fitted model and one importance measure.

B. Implications for Early Warning

The strongest variables, second semester results, are available only at the end of the first year of study. This means that the model using 36 features is not a predictor at enrollment time. Rather, this may be regarded as a warning at the end of the first year. In Table VIII, it is evident what is lost in case there are no such features for the second semester: ROC-AUC and F1-score of XGBoost drop from 0.9241 to 0.9037 and from 0.8077 to 0.7722, and in case there are only enrollment-time features, to 0.8182 and 0.6497. In addition, the threshold should be determined by the institution. At $\tau^* = 0.53$, the model missed 311 dropouts and raised 218 false alarms. Fig. 3 indicates that the use of a lower threshold will identify more dropouts at the risk of raising more false alarms. A threshold optimized for recall may be preferable to the F1-optimal one if support capacity allows. Since class weighting alters the probabilities, the thresholds from Table III cannot be seen as calibrated probabilities of risk.

C. Limitations

There are a number of limitations in this study. First, the threshold values have been determined using the same pooled out-of-fold predictions as those used to calculate accuracy, precision, recall, and the F1-score. This is selection bias in a mild form [21], [22], so these metrics are optimistic. The nested check of XGBoost in Table VI indicates that this selection effect is small: the F1-score decrease is 0.0002 for the main split and 0.0037 averaged across six splits. This effect is not equal to zero and the nested value was not calculated for the thesis table itself. The ROC-AUC value is not related to the threshold and is therefore not affected. Second, the outcome variable had three levels, which were aggregated into two; this means that nothing can be said about distinguishing between students who are enrolled and those who graduated. Third, only a single dataset from a single institution has been used here, so the results cannot be generalized to other universities without new data. Fourth, the confidence intervals from Table V are estimated by resampling the evaluation predictions of one cross-validation run, therefore, they do not include the variability of the model due to retraining. There was no repeated cross-validation of the main table and there was no formal significance testing performed; thus, the slight differences between XGBoost, RF, and LR must not be interpreted as statistically significant.

Fifth, there was no assessment of probability calibration. Sixth, gain importance describes the fitted model rather than causal effects and can be biased towards variables with high cardinality [29]. Permutation and SHAP importance measures were included in the re-run [30]-[32]; however, they both reflect characteristics of the trained model, can split the responsibility between correlated variables, and provide different rankings (Table VII). Furthermore, the thesis did not store its hyperparameters, hence the re-run setup was obtained by matching the thesis table. Table III contains the original results while the re-run reproduces them only approximately for XGBoost, RF, and DT models, and the matching process itself uses the evaluation data. Lastly, the only approaches applied for dealing with class imbalance in this research were class weighting for the five models that support it and threshold tuning, while sampling techniques, such as SMOTE [15], were not compared. Taking these points into account, along with making the code and fold allocations publicly available, would improve the benchmark.

VI. Conclusion

The paper presented a benchmark of seven classical classifiers applied to the publicly available UCI dataset 'Predict Students' Dropout and Academic Success', which is considered in terms of dropout vs. non-dropout for 4,424 students. Under stratified cross-validation with class weighting where applicable and F1-optimal threshold settings, XGBoost showed the best performance, obtaining accuracy of 0.8804, ROC-AUC of 0.9242, and F1 score of 0.8076 with threshold 0.53. RF and LR followed closely, whereas KNN, NB, and DT in particular performed clearly worse. With the threshold chosen, XGBoost identified 78.1% of the dropouts with a precision of 83.6%. The top-ranked features were the approved curricular units and the tuition fee status, which corroborates previous findings with the same dataset. Permutation and SHAP importance revealed the same two leading features, but different ranks for the rest. A re-run with fixed seeds and in-fold preprocessing reproduced the primary table within 0.0012, bootstrapping revealed that XGBoost, RF, and LR cannot be reliably distinguished, and nested threshold selection decreased the F1-score of XGBoost by 0.0037 on average for six splits. The ROC-AUC of XGBoost without second-semester features was 0.9037, and with enrollment-time features only it was 0.8182.

The results are used as a transparent baseline instead of proposing any novel methods. Threshold optimization on the evaluation predictions results in a small residual optimism, which was quantified via nested cross-validation. Future research should incorporate calibration, a comparison of class weighting and resampling techniques, a three-class model, repeated cross-validation with significance tests for the full table, and additional data from other institutions.

FUNDING STATEMENT

The author(s) received no specific funding for this study.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest to report regarding the present study.

AUTHOR CONTRIBUTIONS

Bushra Raiz: conceptualization, methodology, software, formal analysis, and writing of the original draft. Dr. Mudasser Naseer: supervision, and review and editing of the manuscript.

INSTITUTIONAL REVIEW BOARD STATEMENT

Not applicable. This study used only a public, anonymized dataset.

INFORMED CONSENT STATEMENT

Not applicable.

DATA AVAILABILITY STATEMENT

The dataset analyzed in this study is publicly available from the UCI Machine Learning Repository at https://archive.ics.uci.edu/dataset/697 (doi: 10.24432/C5MC89). The survey data of the MS thesis of the first author are analyzed only in a separate companion paper, which has nothing in common with this paper in terms of data, results, tables, and figures.

DECLARATION OF GENERATIVE AI IN SCIENTIFIC WRITING

For preparing this manuscript, the authors have only made use of generative artificial intelligence systems to enhance language proficiency, grammar, and readability.

REFERENCES

[1] L. Kemper *et al.*, "Predicting student dropout: A machine learning approach," *Eur. J. Higher Educ.*, vol. 10, no. 1, pp. 28-47, 2020, doi: 10.1080/21568235.2020.1718520.

[2] I. Sandoval-Palis *et al.*, "Early dropout prediction model: A case study of university leveling course students," *Sustainability*, vol. 12, no. 22, Art. no. 9314, 2020, doi: 10.3390/su12229314.

[3] J. Kabathova and M. Drlik, "Towards predicting student's dropout in university courses using different machine learning techniques," *Appl. Sci.*, vol. 11, no. 7, Art. no. 3130, 2021, doi: 10.3390/app11073130.

[4] A. J. Fernández-García *et al.*, "A real-life machine learning experience for predicting university dropout at different stages using academic data," *IEEE Access*, vol. 9, pp. 133076-133090, 2021, doi: 10.1109/ACCESS.2021.3115851.

[5] M. Quimiz-Moreira *et al.*, "Factors, prediction, explainability, and simulating university dropout through machine learning: A systematic review, 2012-2024," *Computation*, vol. 13, no. 8, Art. no. 198, 2025, doi: 10.3390/computation13080198.

[6] M. Vaarma and H. Li, "Predicting student dropouts with machine learning: An empirical study in Finnish higher education," *Technol. Soc.*, vol. 76, Art. no. 102474, 2024, doi: 10.1016/j.techsoc.2024.102474.

[7] V. Realinho *et al.*, "Predicting student dropout and academic success," *Data*, vol. 7, no. 11, Art. no. 146, 2022, doi: 10.3390/data7110146.

[8] V. Realinho *et al.*, "Predict students' dropout and academic success," UCI Machine Learning Repository, 2021. [Online]. Available: https://archive.ics.uci.edu/dataset/697, doi: 10.24432/C5MC89.

[9] M. V. Martins *et al.*, "Early prediction of student's performance in higher education: A case study," in *Trends and Applications in Information Systems and Technologies* (Advances in Intelligent Systems and Computing, vol. 1365). Cham, Switzerland: Springer, 2021, pp. 166-175, doi: 10.1007/978-3-030-72657-7_16.

[10] S. Romero and X. Liao, "Statistical and machine learning models for predicting university dropout and scholarship impact," *PLOS ONE*, vol. 20, no. 6, Art. no. e0325047, 2025, doi: 10.1371/journal.pone.0325047.

[11] A. Ridwan *et al.*, "Predict students' dropout and academic success with XGBoost," *J. Educ. Comput. Appl.*, vol. 1, no. 2, pp. 1-8, 2024, doi: 10.69693/jeca.v1i2.13.

[12] A. Villar and C. R. V. de Andrade, "Supervised machine learning algorithms for predicting student dropout and academic success: A comparative study," *Discover Artif. Intell.*, vol. 4, Art. no. 2, 2024, doi: 10.1007/s44163-023-00079-z.

[13] J. Niyogisubizo *et al.*, "Predicting student's dropout in university classes using two-layer ensemble machine learning approach: A novel stacked generalization," *Comput. Educ.: Artif. Intell.*, vol. 3, Art. no. 100066, 2022, doi: 10.1016/j.caeai.2022.100066.

[14] S. Kim *et al.*, "Student dropout prediction for university with high precision and recall," *Appl. Sci.*, vol. 13, no. 10, Art. no. 6275, 2023, doi: 10.3390/app13106275.

[15] N. V. Chawla *et al.*, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321-357, 2002, doi: 10.1613/jair.953.

[16] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263-1284, Sep. 2009, doi: 10.1109/TKDE.2008.239.

[17] N. Mduma, "Data balancing techniques for predicting student dropout using machine learning," *Data*, vol. 8, no. 3, Art. no. 49, 2023, doi: 10.3390/data8030049.

[18] J. G. C. Krüger *et al.*, "An explainable machine learning approach for student dropout prediction," *Expert Syst. Appl.*, vol. 233, Art. no. 120933, 2023, doi: 10.1016/j.eswa.2023.120933.

[19] M. Nagy and R. Molontay, "Interpretable dropout prediction: Towards XAI-based personalized intervention," *Int. J. Artif. Intell. Educ.*, vol. 34, no. 2, pp. 274-300, 2024, doi: 10.1007/s40593-023-00331-8.

[20] Z. C. Lipton *et al.*, "Optimal thresholding of classifiers to maximize F1 measure," in *Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2014)* (Lecture Notes in Computer Science, vol. 8725). Berlin, Germany: Springer, 2014, pp. 225-239, doi: 10.1007/978-3-662-44851-9_15.

[21] G. C. Cawley and N. L. C. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *J. Mach. Learn. Res.*, vol. 11, pp. 2079-2107, 2010.

[22] S. Varma and R. Simon, "Bias in error estimation when using cross-validation for model selection," *BMC Bioinformatics*, vol. 7, Art. no. 91, 2006, doi: 10.1186/1471-2105-7-91.

[23] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5-32, 2001, doi: 10.1023/A:1010933404324.

[24] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, San Francisco, CA, USA, 2016, pp. 785-794, doi: 10.1145/2939672.2939785.

[25] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825-2830, 2011.

[26] T. Hastie *et al.*, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009, doi: 10.1007/978-0-387-84858-7.

[27] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861-874, 2006, doi: 10.1016/j.patrec.2005.10.010.

[28] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLOS ONE*, vol. 10, no. 3, Art. no. e0118432, 2015, doi: 10.1371/journal.pone.0118432.

[29] C. Strobl *et al.*, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC Bioinformatics*, vol. 8, Art. no. 25, 2007, doi: 10.1186/1471-2105-8-25.

[30] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer, "Permutation importance: A corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, pp. 1340-1347, 2010, doi: 10.1093/bioinformatics/btq134.

[31] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

[32] S. M. Lundberg *et al.*, "From local explanations to global understanding with explainable AI for trees," *Nat. Mach. Intell.*, vol. 2, no. 1, pp. 56-67, 2020, doi: 10.1038/s42256-019-0138-9.

[33] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York, NY, USA: Chapman & Hall, 1993.