Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 3, 2026

ANTICIPATING STUDENT PERFORMANCE USING FEATURE ENGINEERING AND MACHINE LEARNING CLASSIFIERS

[*1]Haroon Khan, [2]Farah Saqib, [3]Ghulam Irtaza, [4]Adeel Amjad
[*1]Department of Computer Science, National College of Business Administration & Economics(NCBA&E), Multan, Pakistan.
[2]Department of Computer Science, National College of Business Administration & Economics(NCBA&E), Multan, Pakistan.
[3]Department Of Information Sciences University Of Education, Lahore, 54000, Pakistan.
[4]Department of Computer Science, University of Engineering and Technology (UET), Lahore.
[*1]mohharoon01@gmail.com, [2]farahsaqib1@gmail.com, [3]Ghulam.irtaza@ue.edu.pk, [4]adeelamjad2@gmail.com

DOI: https://doi.org/10.5281/zenodo.23464693

Article History
Received on: 21 Feb, 2026
Accepted on: 25 March, 2026
Published on 28 March, 2026

Copyright @Author
Corresponding Author
Haroon Khan

Abstract
Student performance prediction is a core task of Educational Data Mining, yet the contribution of data preparation, as opposed to model choice, is rarely isolated. This study quantifies how label encoding, one-hot encoding and feature selection change the accuracy of six classifiers, Decision Tree, k-Nearest Neighbor, Support Vector Machine, Artificial Neural Network, Random Forest and Extra Trees, and of a voting ensemble on the x-API dataset (480 students, 16 attributes), which was obtained from Kaggle. Three settings are compared: raw features with three grade classes, engineered features (72 attributes after one-hot encoding) with three classes, and a binary pass/fail task with 20 selected features. Information gain, gain ratio, the Gini index and Pearson correlation identified student absence days, visited resources and raised hands as the strongest predictors. With raw features, Random Forest reached 79% accuracy and the voting ensemble 82.6%. After feature engineering, Random Forest reached 90% (+11 points) and the Support Vector Machine improved from 43% to 75%. In the binary setting, Random Forest, Extra Trees, the neural network and the voting ensemble each reached 98.6%. Compared with previously reported x-API results, including a 2024 study, the engineered models are competitive; because merging grade bands makes the task easier, binary results are reported separately from multi-class results. The findings show that data representation produces larger gains than ensembling and that a pass/fail formulation is well suited to early-warning use.

Keywords: Student performance prediction; One-hot encoding; Feature selection; Binary classification; x-API dataset; Random Forest; Educational data mining

1. Introduction

Educational Data Mining (EDM) applies data mining and machine learning to data generated in educational settings, and the prediction of student performance is one of its core tasks (Romero & Ventura, 2013). The x-API dataset, collected from the Kalboard 360 learning management system and distributed through Kaggle, has become a widely used benchmark for this task because it combines demographic, academic-background and behavioral attributes (Amrieh et al., 2015, 2016). Amrieh et al. (2016) showed that behavioral features improved accuracy by up to 22.1% and that ensemble methods added improvements of up to 25.8%.

Most studies on this benchmark compare algorithms, and the effect of data preparation on accuracy is seldom measured separately. Recent work has begun to address preparation: Farhood et al. (2024) one-hot encoded nominal variables and examined Lasso feature selection on a pass/fail target. The present study extends this line by contrasting three settings on the same dataset, namely raw features, engineered features and a binary task, and by comparing the results systematically with previously published values.

Three terms in the title are used in a specific sense. *One-hot encoding* replaces a categorical variable by one binary indicator per category, so that classifiers do not read arbitrary integer codes as ordered quantities. *Feature selection* keeps the attributes that carry information about the target and removes redundant or uninformative ones (Domingos, 2012; Nargesian et al., 2017). *Binary classification* here means predicting only whether a student passes or fails, instead of assigning one of three grade bands.

The main contributions are as follows:

- A side-by-side evaluation of raw, one-hot-engineered and binary settings of the x-API task using six classifiers and a voting ensemble.
- A feature analysis that combines information gain, gain ratio, the Gini index and Pearson correlation.
- A short, explicit treatment of the binary pass/fail formulation and of why its accuracy is higher.
- A systematic comparison with previous x-API results.

The remainder of the paper is organized as follows. Section 2 reviews related work, Section 3 describes data and methods, Section 4 reports results and the comparison with previous studies, Section 5 discusses the findings, Section 6 states limitations, and Section 7 concludes.

2. Related Work

2.1 The x-API dataset and behavioral features

Amrieh et al. (2015, 2016) introduced behavioral features such as raised hands, visited resources, announcements viewed and discussion participation, collected through the experience API (xAPI). They evaluated Artificial Neural Network, Naïve Bayes and Decision Tree classifiers and applied bagging, boosting and Random Forest, reporting accuracy above 80% when the model was tested on newcomer students. Bendangnuksung and Prabu (2018) later applied a deep neural network to the same data and reported 84.3% accuracy, as summarized by Tang et al. (2024).

The dataset records 16 attributes per student: demographic and background variables, academic-stage variables and four behavioral counts that reflect activity in the learning management system. Because the behavioral counts are collected automatically during the course, they are available before final examinations, which makes the dataset suitable for early prediction. At the same time, the data come from a single platform, a single time window and 480 students, so results obtained on them should be treated as benchmark evidence rather than as general laws.

2.2 Encoding, feature selection and ensembles

Feature engineering builds suitable features from the available ones and removes redundant attributes to improve predictive performance (Nargesian et al., 2017; Khurana et al., 2016). Ensemble methods combine several models and usually give more precise solutions than a single model (Xiao et al., 2018), and Random Forest, which averages many decision trees, remains robust even when variables outnumber observations (Biau & Scornet, 2016).

Several encoding strategies exist for categorical attributes. Label (ordinal) encoding maps categories to integers and is appropriate when the categories have a natural order; one-hot encoding creates one binary column per category and is appropriate for

nominal attributes; frequency and target encodings summarize categories with statistics but require care to avoid leakage. One-hot encoding is the most widely used default for nominal variables, but it increases dimensionality, which can affect distance-based learners and increase the risk of over-fitting on small samples. Feature selection is therefore commonly combined with encoding (Domingos, 2012; Nargesian et al., 2017), which is the design used here.

2.3 Previous Studies

Farhood et al. (2024) compared seven machine-learning and three deep-learning models on the Math and x-API datasets using both k-fold cross-validation and holdout validation. They one-hot encoded the nominal variables (16 to 38 variables for x-API, with the four most frequent categories kept for three high-cardinality variables) and defined the target as Pass for the Middle and High classes and Fail for the Low class. Random Forest achieved the highest mean accuracy among the machine-learning models (92.69% on x-API under holdout validation), and Lasso selection raised the accuracy of k-nearest neighbors from 86.10% to 90.23%. Beckham et al. (2023) used Pearson correlation to identify factors affecting grades, Pallathadka et al. (2023) compared Naïve Bayes, ID3, C4.5 and SVM, Asselman et al. (2023) applied XGBoost, and Tang et al. (2024) combined feature ranking with a weighted-voting ensemble of deep belief networks on questionnaire data.

Table 1. ***Selected studies on student performance prediction.***

Study	Data	Method	Key finding
Amrieh et al. (2016)	x-API	ANN, Naïve Bayes, Decision Tree; bagging, boosting, RF	Behavioral features: up to +22.1% accuracy; ensembles: up to +25.8%; above 80% on newcomers
Bendangnuksung & Prabu (2018)*	x-API	Deep neural network	84.3% accuracy
Pallathadka et al. (2023)	UCI student data	Naïve Bayes, ID3, C4.5, SVM	Compared accuracy and error rate
Beckham et al. (2023)	Student data	Pearson correlation; several ML models	Prior failures r = −0.36; mother's education r = 0.22
Asselman et al. (2023)	Student data	XGBoost	Improved prediction with XGBoost
Farhood et al. (2024)	Math and x-API	7 ML and 3 DL models; one-hot; Lasso	RF 92.69% on x-API (pass/fail)
Tang et al. (2024)	Questionnaire, 628 students	DBN ensemble; Relief and mRMR	RMSE 1.66; R^2 0.743

2.4 Research gap and novelty

Previous x-API studies introduced features or compared algorithms, and the most recent comparison (Farhood et al., 2024) studied one target definition. What remains missing is a direct comparison of the same classifiers on the same data under raw, encoded and binary settings, together with a clear account of how these results sit relative to published values. This paper provides that comparison, which is its principal novelty.

The paper is organized around three research questions. **RQ1:** how much do one-hot encoding and feature selection improve the accuracy of common classifiers relative to raw features? **RQ2:** how do the results differ between a three-class and a binary pass/fail formulation? **RQ3:** how do the results compare with values reported in previous studies on the same dataset?

3. Materials and Methods

3.1 Dataset

The x-API dataset was downloaded from Kaggle (Aljarah, 2016). It contains 480 student records with 16 predictor attributes and one class label, and has no missing values. The sample comprises 305 male and 175 female students, 245 records from the first semester and 235 from the second, and 289 students absent fewer than 7 days against 191 absent for more. Attributes fall into three groups (Table 2): demographic, academic

background, and behavioral. The four behavioral counts are numeric on a 0–100 scale.

The target has three classes: Low (scores 0–69, n = 125, 26%), Medium (70–89, n = 211, 44%) and High (90–100, n = 144, 30%). The classes are moderately imbalanced (Figure 1), which is why precision, recall and F1-score are reported alongside accuracy.

Table 2. ***Attributes of the x-API dataset.***

Group	Attributes
Demographic	Gender; nationality; place of birth (14 countries); parent responsible (father, mother); parent answering survey (yes, no); parent school satisfaction (good, bad)
Academic background	Educational stage (lower level, middle school, high school); grade ID (G-1 to G-12); section (A, B, C); topic (12 courses); semester (first, second)
Behavioral	Raised hands; visited resources; viewing announcements; discussion groups (all 0–100); student absence days (under 7, above 7)

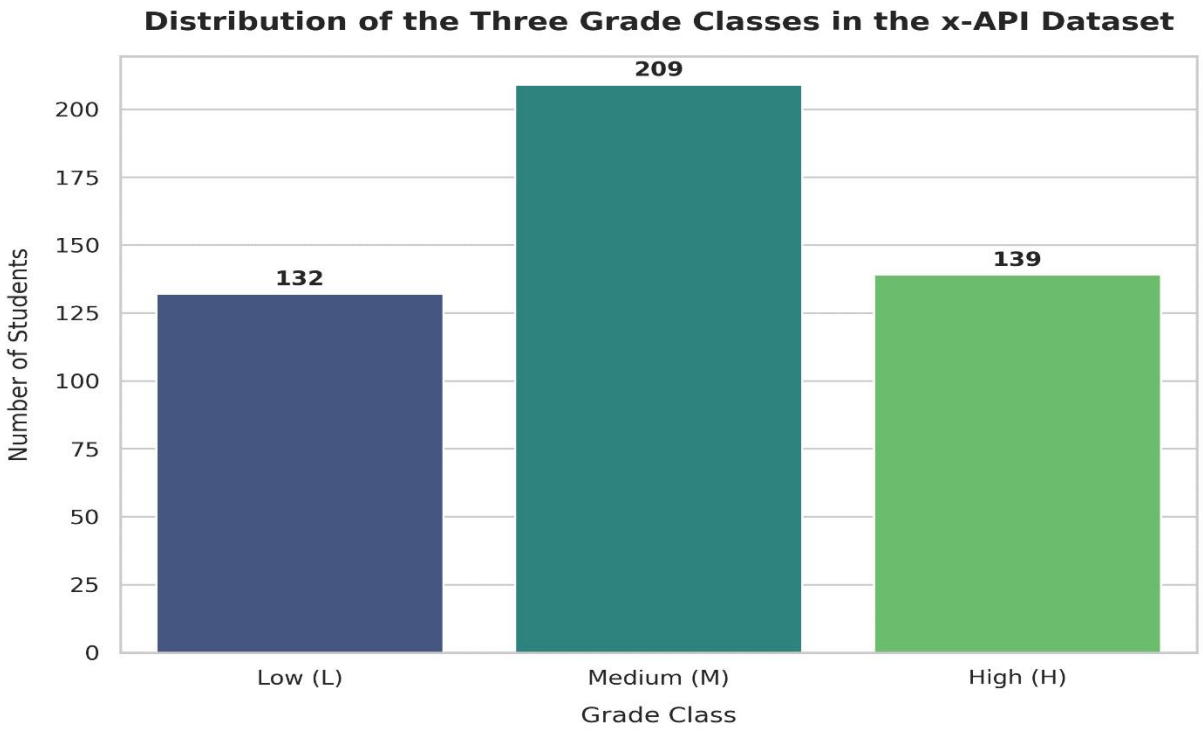


Figure 1. Distribution of the three grade classes in the x-API dataset.

3.2 Experimental design

The same classifiers were execute on three different settings describe in (Figure 2), to study the impact of data representation and feature engineering to effect the accuracy of different algorithms.

(i) **Raw Features (All Features):** Using the original x-API dataset means without change in any feature. This setting serves as the performance baseline.

(ii) **Engineered Features:** Using all 72 attributes generated after applying label encoding, one-hot encoding and normalization of feature, while predicting the same dependent variable name class having target values, $(\{Low\}, \{Medium\}, \{High\})$. This technique tells the effect of feature engineering.

(iii) **Binary Class:** using only 20 selected attributes examine a binary target variable, $\{Fail\}\ or \{Pass\}$ where $\{Low\}$ is marked a $\$Fail$, and $\{Medium\} and\ \{High\}$ are marked as, $\{Pass\}$.

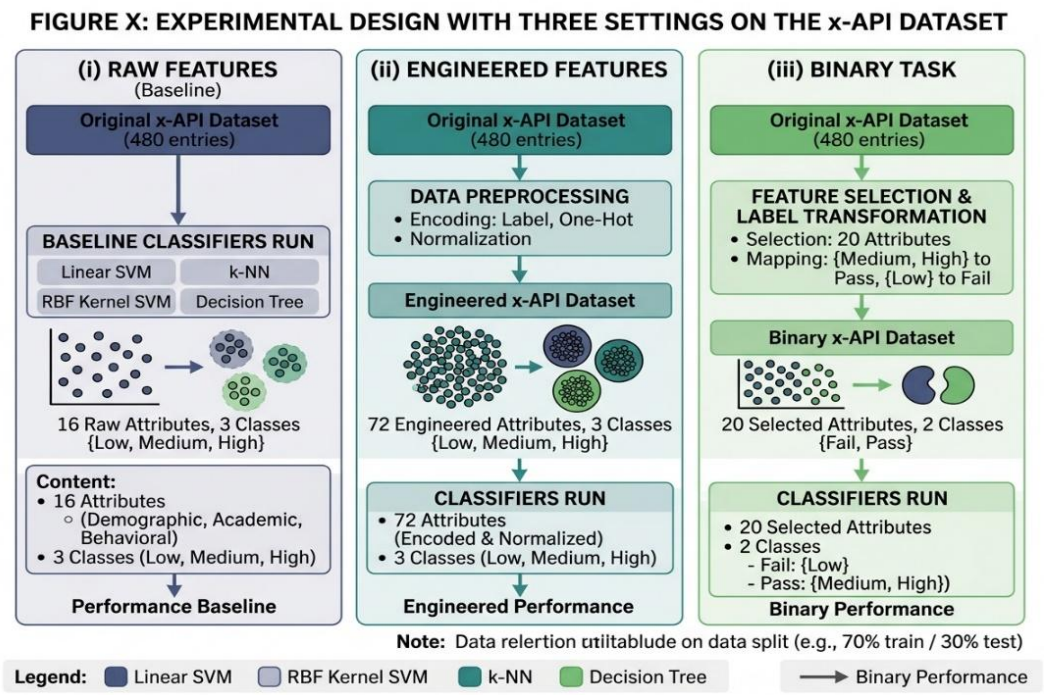


Figure 2. Experimental design with three settings on the x-API dataset.

3.3 Preprocessing and feature engineering

Categorical attributes were label-encoded, which maps each value to an integer (for example, gender F = 1 and M = 2) and allows correlation analysis. Because integer codes can imply an order that does not exist for nominal attributes such as nationality or topic, attributes were also one-hot encoded; this expanded the 16 raw attributes to 72 features. Tree-based models handle categorical values well, but most other algorithms work better with numeric input. Attributes were ranked by information gain (the reduction of class uncertainty once the attribute is known), gain ratio (information gain corrected for its bias toward attributes with many values) and the Gini index (the reduction of class impurity), using the Orange Rank widget, and the Pearson correlation between each encoded attribute and the encoded class was computed in SPSS. Feature selection for the binary task was guided by these results and retained 20 attributes.

To illustrate, an attribute with 12 course names is replaced by 12 binary indicators, and an attribute with 14 countries of birth by 14 indicators, so a record with 16 attributes grows into a much wider representation (72 features in the engineered setting). More columns increase model complexity: with only 480 records, wide representations can promote over-fitting in flexible models, which is one reason feature selection is applied. Normalization rescales numeric attributes, such as the 0–100 behavioral counts, to a common range; this matters for distance-based and gradient-based learners such as KNN, SVM and the neural network and is irrelevant for tree-based models.

3.4 Binary (pass/fail) Class

In the binary setting the three-level class label is collapsed into two outcomes, Pass and Fail. This formulation answers the question that matters most for early warning, namely whether a student is likely to succeed, and it is easier than the three-class task for two reasons. First, it removes confusion between neighboring grade bands, where most misclassifications of a three-class problem occur. Second, with only two classes the accuracy expected by chance is higher. Binary and multi-class accuracies are therefore reported separately and are not ranked against each other. The binary task also used the reduced 20-attribute feature set.

Consider students near a band boundary, for example with scores close to 69 and 70. In a three-class task they belong to different classes and are easily confused, whereas in a binary task the pair is separated only if the pass/fail threshold lies between them. In general, merging classes removes

the decision boundaries that are hardest to learn. The binary results therefore indicate how well a model separates students who are likely to succeed from those at risk, not how well it forecasts grade bands, and any comparison with three-class accuracy should be made with this difference in mind.

3.5 Classifiers, setup and evaluation

Six classifiers were used: Decision Tree (DT), k-Nearest Neighbor (KNN; Deng et al., 2016), Support Vector Machine (SVM; Guenther & Schonlau, 2016), Artificial Neural Network (ANN, a multilayer perceptron), Random Forest (RF; Biau & Scornet, 2016) and Extra Trees (ET), which averages randomized trees fitted on sub-samples. A voting (average) ensemble combined the predictions of the individually trained classifiers. Experiments were run with Orange 3 and Python (Spyder, Python 3.6, Anaconda) on an Intel Core i3 PC (2.00 GHz, 4 GB RAM), with training and testing splits of 70:30 and 80:20. Performance was measured with accuracy, precision, recall and F1-score derived from the confusion matrix (Fawcett, 2006; Powers, 2020):

Decision Tree. A decision tree partitions the data by a sequence of attribute tests chosen to maximize class purity; it is easy to interpret but prone to over-fitting. **KNN.** KNN assigns the class that is most common among the k closest training records; because it relies on distances, it is sensitive to feature scaling and can lose accuracy as the number of sparse dimensions grows. **SVM.** The SVM seeks the maximum-margin boundary between classes and benefits from scaled inputs. **ANN.** The multilayer perceptron learns non-linear combinations of inputs through back-propagation and needs scaled inputs and sufficient data. **Random Forest and Extra Trees.** Both average many randomized decision trees: Random Forest uses bootstrap samples and random attribute subsets, and Extra Trees additionally chooses split thresholds at random, which reduces variance. **Voting ensemble.** The predictions of the individual classifiers are combined by voting (average method). The influence of hyperparameter settings was not examined in this study.

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

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

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

$$F1 = 2 \times (Precision \times Recall) / (Precision + Recall)$$

Precision indicates how many of the records assigned to a class truly belong to it, recall how many records of a class were found, and F1 balances the two. In multi-class tasks the measures are defined per class and summarized into a single value; because the summary can differ from accuracy when classes are unbalanced, both are reported.

4. Results

4.1 Feature analysis

Student absence days was the most informative attribute under all three criteria (information gain 0.397, gain ratio 0.410, Gini 0.131), followed by visited resources (0.391) and raised hands (0.362); semester, stage and section carried almost no information (Figure 3). Pearson correlation confirmed this pattern (Table 3). Correlations with the class were significant ($p < .05$) for parent responsible, raised hands, absence days, visited resources, parent school satisfaction, announcements viewed, gender, parent survey participation, discussion, topic and place of birth, whereas nationality, stage, grade and section were not significant. The signs depend on the integer coding of the class label and are interpreted as association strength only.

	#	Info. gain	Gain ratio	Gini
StudentAbsenceDays	2	0.397	0.410	0.131
VisITedResources		0.391	0.195	0.145
raisedhands		0.362	0.181	0.139
AnnouncementsView		0.253	0.127	0.098
ParentAnsweringSurvey	2	0.150	0.152	0.055
NationalITy	14	0.128	0.052	0.045
Relation	2	0.126	0.129	0.049
PlaceofBirth	14	0.123	0.051	0.046
ParentschoolSatisfaction	2	0.107	0.111	0.040
Discussion		0.088	0.044	0.038
Topic	12	0.076	0.023	0.030
gender	2	0.052	0.055	0.019
GradeID	10	0.047	0.019	0.019
Semester	2	0.012	0.012	0.005
StageID	3	0.011	0.008	0.005
SectionID	3	0.007	0.006	0.003

Figure 3. Ranking of Features.

Table 3. ***Significant Pearson correlations with the class label (N = 480).***

Attribute	r	p
Parent responsible	−.272	< .001
Raised hands	−.231	< .001
Student absence days	−.199	< .001
Visited resources	−.162	< .001
Parent school satisfaction	−.146	.001
Announcements viewed	−.144	.002
Gender	.124	.007
Parent answering survey	−.123	.007
Discussion	−.122	.007
Topic	.104	.023

Place of birth	−.092	.044

4.2 Raw Features (All Features)

With 16 raw attributes, Random Forest was the best individual model (79%), followed by Extra Trees (78.5%), the neural network (73.6%), Decision Tree (66%), KNN (61%) and SVM (43%). The SVM combined a precision of 73% with a recall of 43% and an F-score of 31%, indicating that it did not separate the three classes. The voting ensemble reached 82.6%, better than every individual model (Table 4, Figure 4).

4.3 Engineered Features

After encoding and normalization (72 features), every classifier improved (Table 5). Random Forest rose from 79% to 90%, Extra Trees from 78.5% to 89%, Decision Tree from 66% to 83%, SVM from 43% to 75%, KNN from 61% to 70% and the neural network from 73.6% to 78.4%. The voting ensemble rose from 82.6% to 89% but no longer exceeded Random Forest (90%), because the strong tree-based learners already captured most of the available signal.

The size of the gain depended on the starting point. The models that were weakest on raw features gained the most (SVM +32 points, Decision Tree +17), the strongest raw models gained 10–11 points (Random Forest, Extra Trees), and the neural network gained least (+4.8). KNN improved by 9 points but remained the lowest-scoring model (70%), which is consistent with the sensitivity of distance-based methods to the wider, sparser representation created by one-hot encoding.

4.4 Binary Class Pass/Fail with Selected Features

With 20 selected attributes and two classes, Random Forest, Extra Trees, the neural network and the voting ensemble each reached 98.6% accuracy, followed by Decision Tree (96.5%), KNN (94%) and SVM (91%).

The test sets are small, which limits how finely models can be separated. If a 30% test split of the 480 records (144 records) was used, an accuracy of 98.6% corresponds to 142 correct predictions, that is, two errors, and a single record changes accuracy by about 0.7 points. The tie among Random Forest, Extra Trees, the neural network and the voting ensemble therefore indicates that all four separate pass from fail almost perfectly on this split, not that they are equivalent in general.

Table 4. ***Performance on the x-API dataset (Acc = accuracy, P = precision, R = recall, F = F1; %).***

Setting	Classifier	Features	Acc	P	R	F
All Features	Random Forest	16	79	81	79	79
	Extra Trees	16	78.5	80	78	79
	Neural Network	16	73.6	74	74	74
	Decision Tree	16	66	67	67	67
	k-NN	16	61	64	61	61
	SVM	16	43	73	43	31
	Voting ensemble	16	82.6	83	83	83
Engineered Features	Random Forest	72	90	91	90	90
	Extra Trees	72	89	91	89	89
	Neural Network	72	78.4	81	78	79
	Decision Tree	72	83	85	83	83

	k-NN	72	70	72	69	69
	SVM	72	75	85	75	74
	Voting ensemble	72	89	89	89	89
	Random Forest	20	98.6	99	99	99
	Extra Trees	20	98.6	99	99	99
	Neural Network	20	98.6	98	98	98
Selected Features to predict Binary Class	Decision Tree	20	96.5	97	97	97
	k-NN	20	94	94	94	94
	SVM	20	91	91	91	91
	Voting ensemble	20	98.6	99	99	99

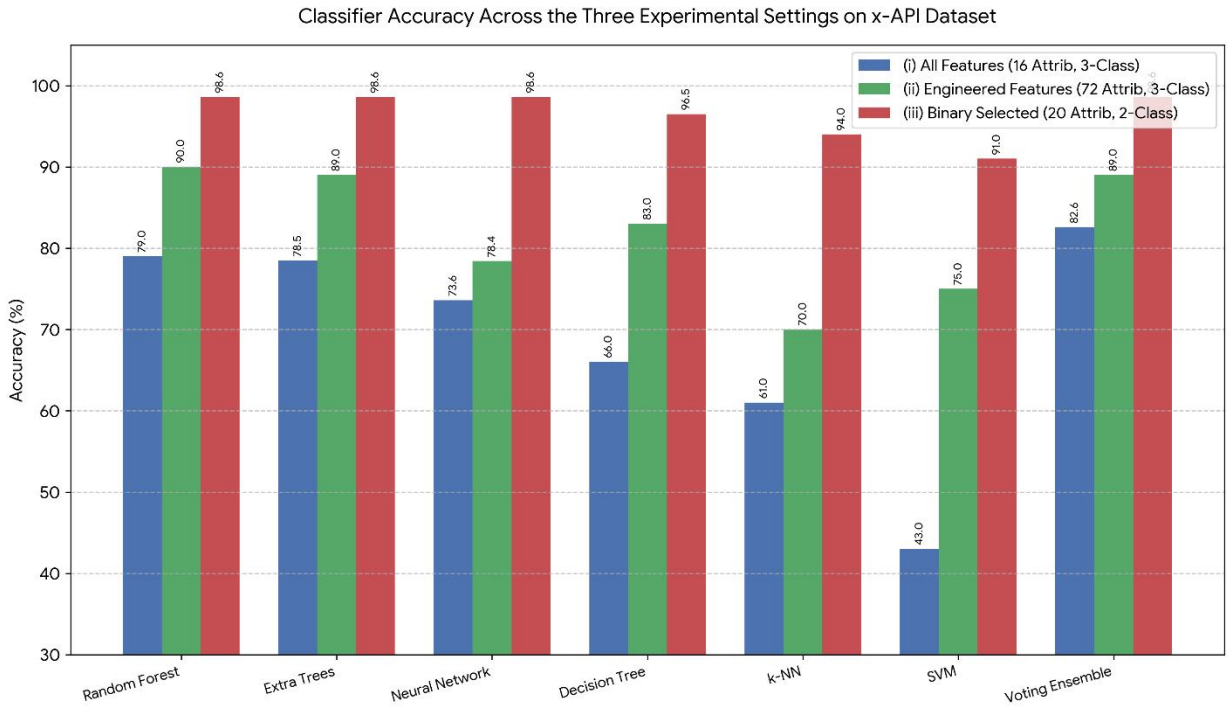


Figure 4. Accuracy of the classifiers on the x-API dataset under the three settings.

Table 5. ***Accuracy change from feature engineering on the three-class task (%).***

Classifier	Raw (16 features)	Engineered (72 features)	Change (points)
Random Forest	79	90	+11
Extra Trees	78.5	89	+10.5
Decision Tree	66	83	+17
k-NN	61	70	+9
SVM	43	75	+32

Neural Network	73.6	78.4	+4.8
Voting ensemble	82.6	89	+6.4

4.5 Does encoding give the best result?

For the three-class task, yes: all seven models improved after encoding, and the best single model rose from 79% to 90%. The highest accuracy overall (98.6%) occurred in the binary setting, but this setting combines three changes at once: encoding, feature selection and a simpler two-class label. Because no binary experiment on raw features was run, the part of the 98.6% attributable to encoding alone cannot be separated from the part due to class reduction and selection. The three-class results (79% to 90%) are therefore the cleanest evidence of the effect of encoding.

4.6 F1-score and error profile

Accuracy can overstate performance when classes are unbalanced, so Table 6 and Figure 5 report the F1-score and its gap to accuracy. In almost every cell the two measures agree within one point, which indicates that errors are not concentrated in a single class. The exception is the SVM on raw features, where accuracy (43%) exceeds the F1-score (31%) by 12 points. This is consistent with a model that predicts mostly the majority class: the Medium class holds 44% of the records, close to the SVM's accuracy. Encoding removed the problem (75% accuracy, 74% F1). The F1-scores of the voting ensemble followed its accuracy in every setting.

Table 6. *Accuracy minus F1-score on the x-API dataset (percentage points).*

Classifier	Raw, 3 classes	Engineered, 3 classes	Binary
Random Forest	0.0	0.0	−0.4
Extra Trees	−0.5	0.0	−0.4
Neural Network	−0.4	−0.6	0.6
Decision Tree	−1.0	0.0	−0.5
k-NN	0.0	1.0	0.0
SVM	12.0	1.0	0.0
Voting ensemble	−0.4	0.0	−0.4

IEER Institute for Excellence in Education & Research

4.7 Comparison with Previous Studies

Table 9 and Figure 6 place the results beside values reported for the x-API data. The comparison is indicative, because protocols differ. Three points distinguish the present study from earlier work. First, unlike Amrieh et al. (2016), whose contribution was the introduction of behavioral features, this study isolates the effect of data representation. Second, Farhood et al. (2024) one-hot encoded nominal variables and evaluated Lasso selection on one pass/fail target; here raw, encoded and binary settings are contrasted on the same data and the ranking is information-theoretic. Third, a voting ensemble is evaluated in every setting, which shows when ensembling helps (raw features, +3.6 points over Random Forest) and when it does not (engineered features, −1 point).

In the three-class setting, the engineered Random Forest (90%) exceeds the 84.3% reported for a deep neural network (Bendangnuksung & Prabu, 2018) and the above-80% newcomer accuracy of Amrieh et al. (2016). In the binary setting, the models here (98.6%) exceed the 92.69% of Random Forest and 91.18% of the gradient-boosted neural network reported by Farhood et al. (2024). This difference should not be read as superiority: Farhood et al. defined Fail as the Low class and Pass as Medium or High, the feature sets differ (20 selected attributes here), and their protocol used 100 repeated evaluations whereas this study used single hold-out splits.

Table 9. *Results on the x-API dataset compared with previous studies.*

Study	Setting	Method	Reported result
Amrieh et al. (2016)	Multi-class, behavioral features	Classifiers with bagging, boosting, RF	Above 80% on newcomer students; up to +25.8% from ensembles
Bendangnuksung & x-API		Deep neural network	84.3%

Prabu (2018)*			
Farhood et al. (2024)	Pass/fail, holdout	Random Forest	92.69%
Farhood et al. (2024)	Pass/fail, holdout	Gradient-boosted neural network	91.18%
Farhood et al. (2024)	Pass/fail, with Lasso	k-NN	86.10% to 90.23%
This study	3 classes, raw features	RF; voting ensemble	79%; 82.6%
This study	3 classes, engineered features	RF; voting ensemble	90%; 89%
This study	Binary, 20 selected features	RF, ET, ANN, voting	98.6%

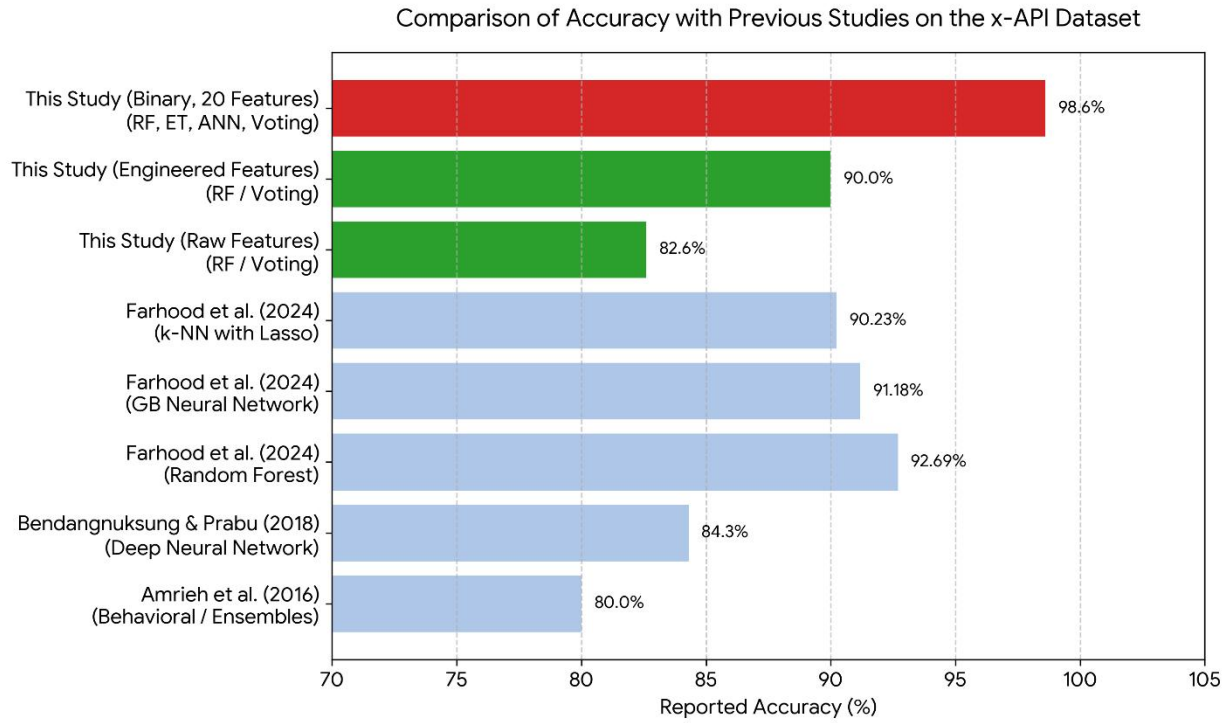


Figure 5. Accuracy reported in previous studies compared with this study.

5. Discussion

Behavior over background. Absence days, resource visits and hand-raising were the most informative attributes under every criterion and were significantly correlated with the class. Engagement during the course is therefore a stronger signal than structural attributes such as stage, section or semester, which is consistent with the behavioral-feature findings of Amrieh et al. (2016) and with the feature importances reported by Farhood et al. (2024).

Why encoding helped. Several classifiers depend on distances or weights and can misread integer codes of nominal attributes as ordered values. The weak SVM result on raw features (43% accuracy, 31% F-score) and its recovery to 75% after one-hot encoding are consistent with this explanation. Tree-based models handle categorical values well and started from the highest raw accuracy, yet still gained about 10 points, so encoding is not only a remedy for distance-based methods.

When ensembling helps. Voting improved on every individual model with raw features, but not on Random Forest after encoding. Ensembling is thus most useful when the base learners are weak or noisy, and its benefit shrinks as the representation improves.

Binary results. The binary formulation removes confusion between adjacent grade bands and raises the chance baseline, so the 98.6% scores describe an easier problem and should not be compared directly with the three-class results. Their value is practical: a pass/fail alert is what an advisor needs to decide whether to intervene.

5.1 Institutional Implications

Implications for practice. An advisor-facing early-warning tool would use the binary model: it needs only attributes that are available during the course (absence days, resource visits, hand-raising) and produces a simple pass/fail alert. The three-class model is more informative but less accurate and is better suited to planning than to alerts. In both cases the output should support, not replace, human judgment.

Generalizability and reproducibility. The x-API data are publicly available, so the experiment can be repeated. Complete reproducibility additionally requires the exact preprocessing steps, the list of the 72 engineered features and of the 20 selected attributes, the random seeds of the splits and the hyperparameter settings, which should be released with the paper. Whether the same pattern holds on other datasets, in particular on locally collected questionnaire data, remains to be tested.

6. Limitations

The dataset is small (480 records) and comes from one learning platform, and the classes are imbalanced. Evaluation used single hold-out splits (70:30 and 80:20) without cross-validation or confidence intervals, and the influence of hyperparameter settings was not examined. The three settings differ in more than one respect in the binary case (label, features), and no binary experiment on raw features was run, so the effect of encoding cannot be isolated there. Comparisons with previous studies are indicative because of differences in protocol and target definition.

6.2 Threats to validity

Threats to validity. Internal validity is affected by the use of single hold-out splits and by the possibility that feature ranking was computed on all records before splitting, which can inflate estimates. Construct validity depends on how the binary label was formed from the three grade bands. External validity is limited to one dataset from one learning platform. Statistical conclusion validity is limited because no confidence intervals or significance tests were computed; with a test set of roughly 144 records, differences of one or two points should not be over-interpreted.

7. Conclusion

This study showed that data representation matters more than the choice of classifier on the x-API task. One-hot encoding and normalization raised Random Forest from 79% to 90% and SVM from 43% to 75%, and a binary pass/fail task with 20 selected features reached 98.6%. Absence days, resource visits and raised hands were the strongest predictors. Ensembling helped with raw features but not after encoding. Future work should add a binary experiment on raw features, repeated cross-validation, hyperparameter tuning and per-class analysis, and should test the pipeline on other datasets.

Future research should include (i) a binary experiment on raw features, which would complete a two-by-two design of label (three-class or binary) and representation (raw or engineered) and isolate the effect of encoding in each case; (ii) repeated stratified cross-validation with confidence intervals and significance tests; (iii) systematic hyperparameter tuning; (iv) alternative encodings and selection methods, including embedded methods such as Lasso; and (v) validation on other datasets, including locally collected questionnaire data.

References

1. Aljarah, I. (2016). *Students' academic performance dataset (xAPI-Edu-Data)* [Data set]. Kaggle. https://www.kaggle.com/datasets/aljarah/xAPI-Edu-Data
2. Amrieh, E. A., Hamtini, T., & Aljarah, I. (2015). Preprocessing and analyzing educational data set using X-API for improving student's performance. *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, 1–5.
3. Amrieh, E. A., Hamtini, T., & Aljarah, I. (2016). Mining educational data to predict student's academic performance using

ensemble methods. *International Journal of Database Theory and Application, 9*(8), 119–136.
4. Asselman, A., Khaldi, M., & Aammou, S. (2023). Enhancing the prediction of student performance based on the machine learning XGBoost algorithm. *Interactive Learning Environments, 31*, 3360–3379. https://doi.org/10.1080/10494820.2021.1928235
5. Beckham, N. R., Akeh, L. J., Mitaart, G. N. P., & Moniaga, J. V. (2023). Determining factors that affect student performance using various machine learning methods. *Procedia Computer Science, 216*, 597–603. https://doi.org/10.1016/j.procs.2022.12.174
6. Bendangnuksung, & Prabu, P. (2018). Students' performance prediction using deep neural network. *International Journal of Applied Engineering Research, 13*, 1171–1176.
7. Biau, G., & Scornet, E. (2016). A random forest guided tour. *Test, 25*(2), 197–227.
8. S. U. R. Khan, A. Raza, I. Shahzad and G. Ali, "Enhancing Concrete and Pavement Crack Prediction through Hierarchical Feature Integration with VGG16 and Triple Classifier Ensemble," 2024 Horizons of Information Technology and Engineering (HITE), Lahore, Pakistan, 2024, pp. 1-6. DOI: 10.1109/HITE63532.2024.10777242.
9. Raza, Asif, Salahuddin, & Inzamam Shahzad. (2024). Residual Learning Model-Based Classification of COVID-19 Using Chest Radiographs. Spectrum of Engineering Sciences, 2(3), 367–396.
10. Deng, Z., Zhu, X., Cheng, D., Zong, M., & Zhang, S. (2016). Efficient kNN classification algorithm for big data. *Neurocomputing, 195*, 143–148.
11. Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM, 55*(10), 78–87.
12. Al-Khasawneh, Mahmoud Ahmad, Asif Raza, Saif Ur Rehman Khan, and Zia Khan. "Stock Market Trend Prediction Using Deep Learning Approach." Computational Economics (2024): 1-32
13. Farhood, H., Joudah, I., Beheshti, A., & Muller, S. (2024). Evaluating and enhancing artificial intelligence models for predicting student learning outcomes. *Informatics, 11*(3), 46. https://doi.org/10.3390/informatics11030046
14. Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861–874.
15. Raza, Asif , Soomro, M. H., Shahzad, I., & Batool, S. (2024). Abstractive Text Summarization for Urdu Language. Journal of Computing & Biomedical Informatics, 7(02).
16. Guenther, N., & Schonlau, M. (2016). Support vector machines. *The Stata Journal, 16*(4), 917–937.
17. Khurana, U., Turaga, D., Samulowitz, H., & Parthasarathy, S. (2016). Cognito: Automated feature engineering for supervised learning. *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW).*
18. Bilal, Omair, Arash Hekmat, Inzamam Shahzad, Asif Raza, and Saif Ur Rehman Khan. "Boosting Machine Learning Accuracy for Cardiac Disease Prediction: The Role of Advanced Feature Engineering and Model Optimization." The Review of Socionetwork Strategies (2025): 1-30.
19. Khan, S. R., Asif Raza, Inzamam Shahzad, & Hafiz Muhammad Ijaz. (2024). Deep transfer CNNs models performance evaluation using unbalanced histopathological breast cancer dataset. Lahore Garrison University Research Journal of Computer Science and Information Technology, 8(1).
20. Nargesian, F., Samulowitz, H., Khurana, U., Khalil, E. B., & Turaga, D. S. (2017). Learning feature engineering for classification. *IJCAI.*
21. O. Bilal, Asif Raza, S. ur R. Khan, and Ghazanfar Ali, "A Contemporary Secure Microservices Discovery Architecture with Service Tags for Smart City Infrastructures ", VFAST trans. softw. eng., vol. 12, no. 1, pp. 79–92, Mar. 2024
22. Pallathadka, H., Wenda, A., Ramirez-Asís, E., Asís-López, M., Flores-Albornoz, J., &

Phasinam, K. (2023). Classification and prediction of student performance data using various machine learning algorithms. *Materials Today: Proceedings, 80*, 3782–3785. https://doi.org/10.1016/j.matpr.2021.07.382

23. Asif Raza, Inzamam Shahzad, Ghazanfar Ali, and Muhammad Hanif Soomro. "Use Transfer Learning VGG16, Inception, and Reset50 to Classify IoT Challenge in Security Domain via Dataset Bench Mark." Journal of Innovative Computing and Emerging Technologies 5, no. 1 (2025).
24. Powers, D. M. (2020). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.
25. Khan, S. R., Raza, A., Waqas, M., & Raphay Zia, M. A. (2024). Efficient and Accurate Image Classification via Spatial Pyramid Matching and SURF Sparse Coding. Lahore Garrison University Research Journal of Computer Science and Information Technology, 7(4).
26. Romero, C., & Ventura, S. (2013). Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 3*(1), 12–27.
27. Tang, B., Li, S., & Zhao, C. (2024). Predicting the performance of students using deep ensemble learning. *Journal of Intelligence, 12*(12), 124. https://doi.org/10.3390/jintelligence12120124
28. Xiao, Y., Wu, J., Lin, Z., & Zhao, X. (2018). A deep learning-based multi-model ensemble method for cancer prediction. *Computer Methods and Programs in Biomedicine, 153*, 1–9.

IEER
Institute for Excellence in Education & Research