Predicting student performance in a blended learning environment using learning management system interaction data

The current issue and full text archive of this journal is available on Emerald Insight at:
https://www.emerald.com/insight/2210-8327.htm

Predicting student performance in a blended learning environment using learning management system interaction data

Kiran Fahd
College of Engineering and Science, Victoria University, Melbourne, Australia
Shah Jahan Miah
Newcastle Business School, The University of Newcastle, Newcastle, Australia, and
Khandakar Ahmed
College of Engineering and Science, Victoria University, Melbourne, Australia

Received 17 June 2021
Revised 11 August 2021
13 September 2021
Accepted 15 September 2021

Abstract

Purpose – Student attritions in tertiary educational institutes may play a significant role to achieve core values leading towards strategic mission and financial well-being. Analysis of data generated from student interaction with learning management systems (LMSs) in blended learning (BL) environments may assist with the identification of students at risk of failing, but to what extent this may be possible is unknown. However, existing studies are limited to address the issues at a significant scale.

Design/methodology/approach – This study develops a new approach harnessing applications of machine learning (ML) models on a dataset, that is publicly available, relevant to student attrition to identify potential students at risk. The dataset consists of the data generated by the interaction of students with LMS for their BL environment.

Findings – Identifying students at risk through an innovative approach will promote timely intervention in the learning process, such as for improving student academic progress. To evaluate the performance of the proposed approach, the accuracy is compared with other representational ML methods.

Originality/value – The best ML algorithm random forest with 85% is selected to support educators in implementing various pedagogical practices to improve students' learning.

Keywords Machine learning, Classification, Design research, Higher education, Learning management systems (LMS), LMS data, Student attrition, Student retention, Random forest, Decision tree, Boosting ensemble technique, Student academic performance

Paper type Conceptual paper

1. Introduction

Student attrition [1] is of a great concern to, and an extraordinarily challenging issue to address for higher education (HE) providers. Various factors contribute to student attrition [1, 2], such as withdrawal from courses because of academic failure, peer pressure, financial issues, inter-institutional transfer, employment-related factors or myriad personal reasons.

Academic progress is frequently cited as a key factor associated with student attrition [3], with HE providers offering various interventions to improve it. The objective of one form of intervention, the student support program, is to extend additional, tailored academic support,

Applied Computing and Informatics
Emerald Publishing Limited
e-ISSN: 2210-8327
p-ISSN: 2634-1964
DOI 10.1108/ACI-06-2021-0150

typically involving academic or language services, to students experiencing academic problems. By doing so the student attrition rate is reduced, and the reputation and financial viability of the HE institution are maintained or even improved.

While support programs may address some issues with attrition rates [4], the first step in this process—the identification of at-risk students—is a manual and time-consuming exercise that can be biased by personnel involvement. Moreover, the delay between identification of an at-risk student, the onset of intervention and any assessment of the effect of this intervention, can be lengthy. Early, if not real-time identification of struggling students is preferable because it would enable educators to provide timely and appropriate support to students when it is most needed and effective [5] and almost real-time assessment of the effects of any intervention.

Various techniques have been used to predict student academic progress [6] through the application of different machine learning (ML) algorithms on student demographic, socio-economic, pre-enrollment, enrollment, academic and learning management system (LMS) data [6, 7], the latter automatically generated through student interaction. Approaches to identify these at-risk students based on data related to socioeconomic and cultural factors [4, 8] lack precision. We apply ML algorithms to identify struggling students accurately and rapidly from a dataset collected from student LMS interaction. We investigate how the application of existing ML techniques can more accurately and rapidly identify at-risk students.

After a brief review of related work, we define our research question, explain the dataset that we use and any necessary pre-processing of it, the ML algorithms used for data analysis and the various classification techniques that we employ. We conclude this contribution with an evaluation and brief discussion of results from the tree-based classification algorithms, implications of this research and future research directions.

2. Related works

The application of ML techniques to predict and improve student performance, recommend learning resources and identify students at-risk has increased in recent years. Two main factors affect the identification of students at risk using ML: the dataset and delivery mode and the type of ML algorithm used. We took a stock of recent literature to analyze a wide variety of dataset features, delivery modes and ML techniques for predicting student performance, and the same is presented as supplementary material available at: https://github.com/KFVU/ML-C/blob/b32b95655f13ce4f623e703df6b07b850688d8eb/1.pdf.

2.1 Important attributes for predicting student academic performance

Aspects of a student's demographic and socio-economic background (e.g. place of birth, disability, parent academic and job background, residing region, gender, socioeconomic index, health insurance, frequency of going out with friends (weekday and weekend) and financial status) [4, 8–13], pre-enrollment (e.g. high school or level 12 performance and grades, entrance qualification, SAT scores, English and math grades, awards and the school they attended) [4, 8–10, 14, 15], enrollment (e.g. enrollment date, enrollment test marks, the number of courses students previously enrolled in, type of study program and study mode) [16, 17], tertiary academic (e.g. attendance, number of assessment submissions, student engagement ratio, major, time left to complete the degree, course credits, semester work marks, placements and count and date of attempted exams) [4, 14–18] and LMS-based data have all been studied in previous analyses regarding the prediction of student academic performance.

Student record such as grade point average (GPA) has been frequently used as a categorical variable, as have a semester or final results of a student [4, 8–12, 14–20] and the graduate or drop out the status of a student [16]. These are considered to be significant indicators of academic potential. Therefore, we consider the final result of a student to be the nominal variable on which basis we assess a student's study performance.

Few studies have used LMS-generated data to predict student achievement. Attributes include the frequency of interaction of a student with each module on LMS [21], LMS log data, counts of hits, forum post details, counts of assessments viewed and submitted on LMS [11], start and end dates and assessment submission dates [20]. LMS data are automatically generated and stored by the LMS, which is cost-effective, and the data are accessible and relatively easy to analyze. LMS data provide complete information about a student's engagement in online learning sessions and workshops. Few studies have researched correlations between LMS attributes, selection of relevant attributes and tuning of classifier algorithm parameters for accurate prediction of student progress. To our knowledge, the use of student learning behavior and LMS participation in blended learning (BL) has not been previously investigated. Additionally, the focus of most studies was not on the early detection of at-risk students for the purposes of taking timely action to implement remedial measures to improve their progress.

Most research datasets have been acquired from traditional face-to-face or online classroom settings [19], although several studies have used datasets obtained from BL [11, 21]. The BL approach combines traditional classroom environments with online learning, starting from a 10%–25% digital to 90%–75% classroom ratio, to the reverse situation, a 75%–90% digital component. BL represents a transition from synchronous to asynchronous learning and potentially enriches and extends the opportunities for students to learn in ways that were previously unachievable.

2.2 ML techniques used to predict student academic performance

Most studies have applied supervised (as opposed to unsupervised) ML techniques to predict student academic performance. It is easy to understand a reasoning tree of "if–then" rules, with such decision tree based techniques used frequently to predict student academic progress and identify at-risk ones [22]. Each of decision tree [9, 10, 13, 14, 19], random forest [8, 19, 21], J4 [4, 7, 12, 20], decision stump [20], OneR [20], NBTree [20], ID3 [20], PART [12, 20], Naive Bayes [9, 13], neural networks [9, 16], support vector machine [9, 21], logistic regression [16, 19], ZeroR [20], Prism [12, 20], multi-layer perceptron [4] and K-nearest neighbor [9] have been used to predict student progress. Few studies have used clustering ML techniques [15] to improve learning outcomes, of which K-mean has proven popular for the analysis of student data [9, 11, 14]. Extensive studies like [23, 24] reveal decision tree techniques perform better in predictive modeling than other classifiers when used on student data. Combining classifiers as ensemble techniques may provide a better accurate and generalized ML model because of the voting rule or probabilistic averaging [25]. Several studies have applied ensemble techniques [8, 10, 15] to predict student academic performance.

We apply decision tree based classification methods because of their simplicity, appropriateness and ease of interpretation. We used the frequently used tree-based algorithms on an LMS-based dataset to identify students at-risk, to address our research question – "what is the most feasible ML approach to apply to an LMS-interaction dataset to accurately identify at-risk students?"

3. Method

We source freely available data from the UCI (University of California, Irvine) ML repository [26] which comprises 230,318 data instances built from the recordings of about 112 students' activities and interactions while learning with LMS in six laboratory sessions conducted in a simulated e-learning environment. Data were collected from LMS logs, transformed and cleaned into a format appropriate for public dissemination.

We build a dimensional vector using student LMS interaction data, which is then transformed to include response features. This transformed dataset is then used to build an algorithm to identify students at risk. We use five tree-based classifiers (random forest, J48, NBTree, OneR, decision stump) which use a series of if-then decisions to generate highly accurate, easily interpretable predictions, to predict at-risk students. We then compare the performance of these different classification methods using various metrics (e.g. accuracy, precision, recall and *F*-measure). The dataset is fine-tuned by using a Booster ensemble method on each classification method. Finally, based on accuracy, we identify which classifier is most appropriate for building our algorithm to identify at-risk students.

The dataset consists of multiple comma-separated value (csv) files. One set of csv files contains information regarding sessions and students. Each folder represents a session, and each csv file contains data for a specific student identified by their student Id. Each csv file contains data about all exercises performed by each student in a specific session. Each record comprises information regarding dimensions, the activities a student attempted during a specific session for a specific exercise, the start and end times of the activity and other related features. Two additional files contain the final exam grade of a student and attendance records for each student for each session. A summary of dataset dimensions may be perused as supplementary material at: https://github.com/KFVU/ML-C/blob/b32b95655f13ce4f623e703df6b07b850688d8eb/2.pdf.

Our simulation research methods involve dataset cleaning and pre-processing, application of different classification algorithms to the dataset and selection of the most accurate model to predict student performance. This step-by-step process is depicted in Figure 1. This framework explains the series of rigorous and iterative phases required to develop an innovative educational artifact (predictive model) for predicting student progress based on ML techniques [27].

Step 1: Data collection and feature exploration

UCI data comprising 230,318 data instances based on activities and interactions of 112 students with an e-learning system in six sessions were sourced. Each csv file in this dataset consists of 13 attributes of text data alongside numerical attributes (i.e. Exercise, SessionID, Activity, StudentID, Start-time, Idle-time, End-time, Mouse-wheel, Mouse-click-left, Mouse-wheel-click, Mouse-click-right, Keystroke and Mouse-movement). Additional files contain intermediate and final student marks. All csv files for sessions were combined into a single csv file to transform the mixed attribute dataset into a numerical feature dataset with nine attributes. The transformed dataset was obtained by aggregating the attributes for each student for all sessions using algorithm to create and validate the dimensional vector V and in-detail algorithm is given as supplementary material at: https://github.com/KFVU/ML-C/blob/b32b95655f13ce4f623e703df6b07b850688d8eb/3.pdf.

The algorithm employed in the procedure aimed at building and verifying the correctness of aggregated data in the dimensional vector V. Null, empty, or negative values are removed from the dataset. V is first built using aggregated values of each feature for each student, and the total final marks for students are merged with V using StudentID. This extracts records of students who attended all sessions and the final exam. In theory, V should contain data about the students who attended all sessions that can be verified by attendance data in *logs.txt*. A Boolean attribute DV is created for this rule and a *StudentID* attribute is created to store the StudentID attribute of each row of V. A variable (*totalAttendance*) is computed for all rows of V, the value of which (n) is equal to the total sessions a student attended, which in this study is 6 (i.e. $n = 6$). If this Boolean expression is satisfied, the DV value becomes true; if not it

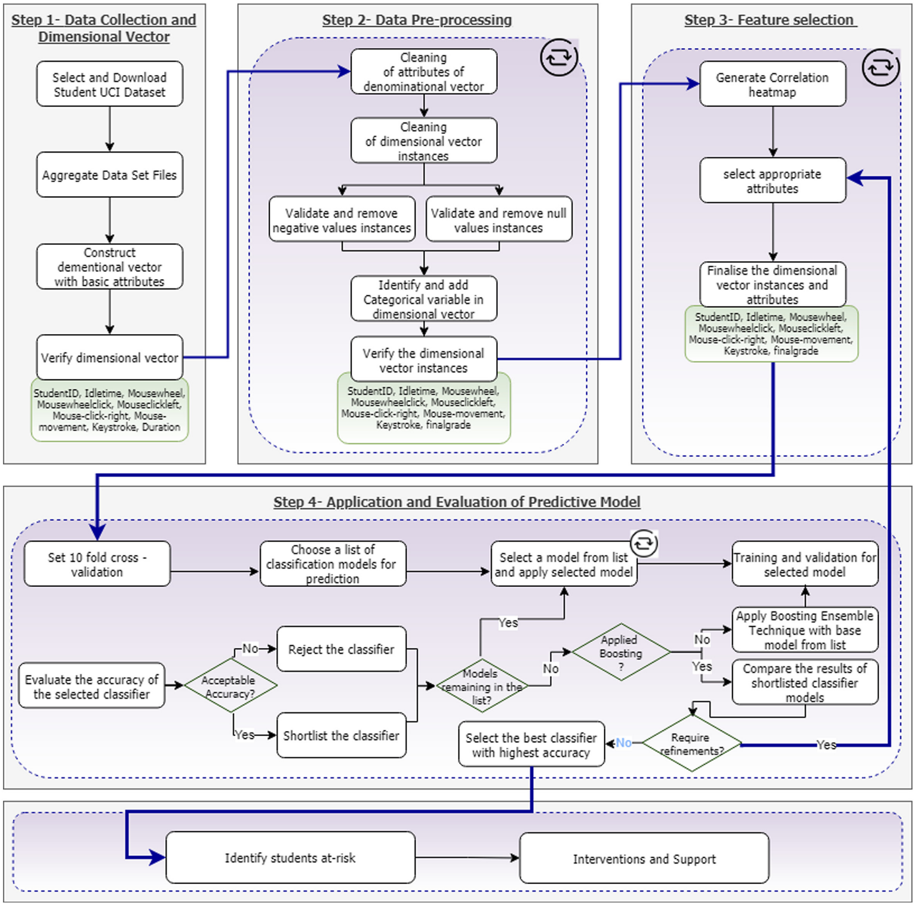

Step 1- Data Collection and Dimensional Vector
Select and Download Student UCI Dataset
Aggregate Data Set Files
Construct dementional vector with basic attributes
Verify dimensional vector
StudentID, Idletime, Mousewheel, Mousewheelclick, Mouseclickleft, Mouse-click-right, Mouse-movement, Keystroke, Duration
Step 2- Data Pre-processing
Cleaning of attributes of denominational vector
Cleaning of dimensional vector instances
Validate and remove negative values instances
Validate and remove null values instances
Identify and add Categorical variable in dimensional vector
Verify the dimensional vector instances
StudentID, Idletime, Mousewheel, Mousewheelclick, Mouseclickleft, Mouse-click-right, Mouse-movement, Keystroke, finalgrade
Step 3- Feature selection
Generate Correlation heatmap
select appropriate attributes
Finalise the dimensional vector instances and attributes
StudentID, Idletime, Mousewheel, Mousewheelclick, Mouseclickleft, Mouse-click-right, Mouse-movement, Keystroke, finalgrade
Step 4- Application and Evaluation of Predictive Model
Set 10 fold cross - validation
Choose a list of classification models for prediction
Select a model from list and apply selected model
Training and validation for selected model
Apply Boosting Ensemble Technique with base model from list
Compare the results of shortlisted classifier models
Applied Boosting ?
No
Yes
Models remaining in the list?
Yes
No
Reject the classifier
Shortlist the classifier
No
Yes
Acceptable Accuracy?
Evaluate the accuracy of the selected classifier
Select the best classifier with highest accuracy
No
Require refinements?
Yes
Identify students at-risk
Interventions and Support

Figure 1.
Proposed methodology using ML to predict student progress

becomes false. This verifies that for each selected instance of the student, the sum of attendance should be equal to the value 6.

Step 2: Dataset pre-processing

Dataset pre-processing involves the cleaning of variables and data instances and converting the dataset into a csv file as an outcome of algorithm 1. After aggregating data instances, the numeric values of final result marks are classified into the categorical variables “Pass” or “Fail,” 62% and 38% of the dataset, respectively.

Step 3: Feature selection

Suitable features are selected in exploratory data analysis, which affects the prediction result. A correlation heatmap is produced using the open-source software Python Pandas is a data analysis and manipulation library; the value sign indicates a +ve or −ve correlation with the final score. For example, if “keystroke” is high or “idle_time” is low then there is a higher probability that the final score is higher. Attribute correlations are depicted using a heatmap (Figure 2).

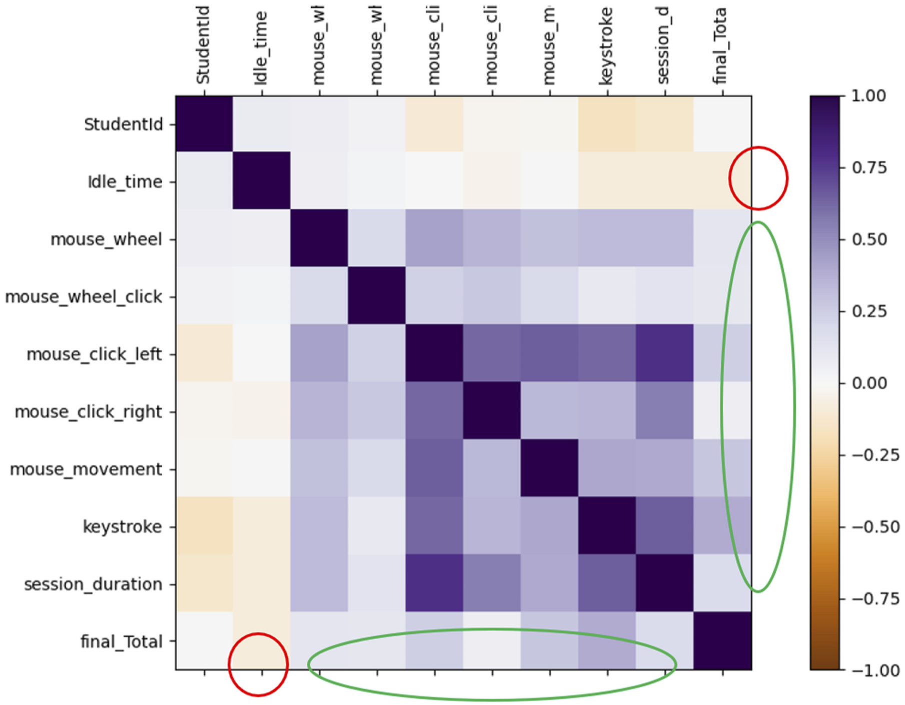


Figure 2.
Heatmap of classified data for ML

Note(s): Red circle shows the negative correlation of Idle_time with the final score and green oval shows the positive correlation of multiple features with the final score

This exploratory analysis supports the selection of features for building classifiers. Because ensemble techniques effectively improve the performance of early prediction models, we use five classifiers to first train our model. An adaptive boosting (AdaBoost) technique is used in the subsequent iteration to improve classification accuracy; this ensemble boosting technique learns from the previous misclassification of data points by increasing their weights and boosting decision trees.

Step 4: Machine learning models—classification algorithm

To undertake the classification of ML techniques we used Waikato Environment, distributed under the GNU General Public License. This workbench offers a wide collection of classification ML algorithms and visualization features. We loaded cleaned and aggregated data into WEKA to apply the classification ML algorithm. We used supervised learning methods to train the model, where the model learns from labeled classes (e.g. Pass, Fail). Random forest, J48, OneR, NBTree and decision stump were used to classify at-risk students.

A 10-fold cross-validation split the student dataset into 10 groups of approximately equal size, wherein the first group was treated as a validation group and the classifier was trained on the nine remaining groups (repeated 10 times). Results for each group are summarized using evaluation scores. Classifier accuracy is presented in Table 1. Classifiers were then tuned with AdaBoost, which sequentially trained several models and combined multiple weak models into a single strong classifier. The tuned classifier was applied to the updated dataset obtained from the former step. The 10-fold cross-validation method is again used to recheck the performance of all classifiers. The accuracy of each ML classifier was used to identify the best of the five (Table 1). Dataset pre-processing, feature selection by exploratory data analysis, ML classifier training and analysis and comparison of performance metrics are iterative steps required to filter attributes and tune the model.

The process of predicting student progress (steps 1 to 4) using the most accurate classification method is then automated by one more algorithm which develops and verifies the dimensional vector V (step I) and an overview of this algorithm may be perused as supplementary material at: https://github.com/KFVU/ML-C/blob/b32b95655f13ce4f623e703df6b07b850688d8eb/4.pdf. The same five ML classifiers (random forest, J48, NBTree, OneR, decision stump) are again applied to V using a k-fold cross validation. The boosting ensemble technique is then applied on V using the five classification algorithms with k-fold cross validation. The accuracy of different ML methods is saved without applying the ensemble technique in vector PM and with it in vector PME. The performance metrics of the five classification models are then compared and the most accurate method is selected. This algorithm fully automates the process of creating the dimensional vector, selecting the best classifier and identifying students with learning difficulties. The process of remedial activities to improve student learning can then commence.

4. Evaluation and discussion

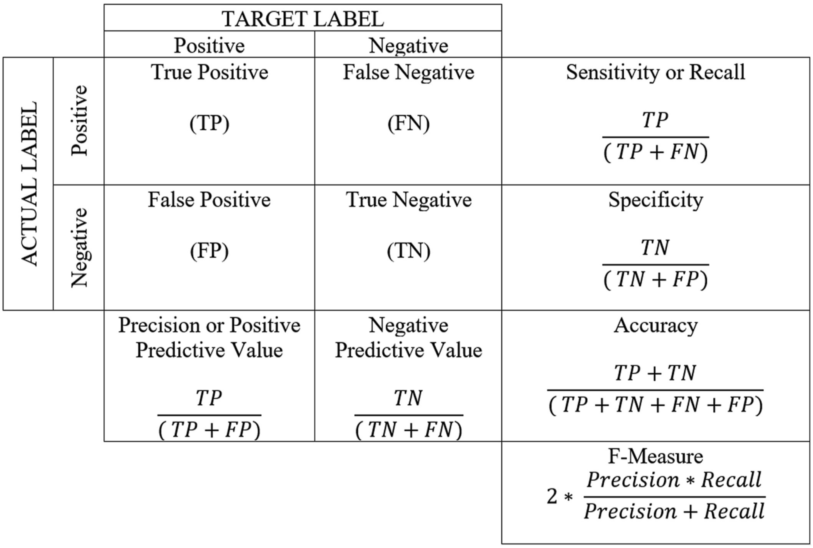
Different performance metrics (classification accuracy, precision, recall, F-measure, root mean square error and incorrectly identified instances) are used to evaluate the five algorithms as presented in Figure 3. Classification Accuracy is used to select the best-performing classifier, which is calculated by using confusion metrics. All five classifiers perform well with high accuracy, demonstrating the feasibility and effectiveness of dataset pre-processing and feature selection. The objective is to maximize TN and minimize FP. Confusion metrics for the five classifiers used to evaluate performance are presented in Figure 4.

To provide timely and appropriate support it is important that our model is accurate. Performance metrics for selected tree-based ML classification algorithms with and without Booster ensemble tuning are presented in Table 1. Classification accuracy (%) represents the ratio of correct classifier prediction over the total number of observations (×100). Random forest with and without ensemble tuning outperforms other classifiers in classification accuracy, precision, recall, F-measure, root mean square error and incorrectly identified

	Accuracy		Precision		Recall		*F*-measure		RMSE		Incorrectly identified instances	
	NE %	*WE* %	*NE* %	*WE* %	*NE* %	*WE* %	*NE* %	*WE* %	*NE* %	*WE* %	*NE* %	*WE* %
Random forest	79.4	85.7	79.3	85.7	79.3	85.7	78.8	84.3	55.1	32.7	20.6	14.2
J48	75	83.7	74.7	82.8	75	83.7	74.6	82.5	48.5	39.4	25	16.3
NBTree	67.4	81.6	47.6	82.2	66.9	81.6	67.4	81.9	67	39.3	32.6	18.3
OneR	61.9	81.6	61.8	81.2	62	81.6	61.9	81.4	61.6	37.8	38.1	18.3
Decision stump	60.9	83.7	60.9	82.2	60.9	83.7	75.7	82.5	45.4	35.3	39.1	16.3

Note(s): *NE* - Evaluation without ensemble method tuning; *WE* - Evaluation with ensemble method tuning

Table 1. Comparison of performance metrics of five ML algorithms



		TARGET LABEL		
		Positive	Negative	
ACTUAL LABEL	Positive	True Positive (TP)	False Negative (FN)	Sensitivity or Recall $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP)	True Negative (TN)	Specificity $\frac{TN}{(TN+FP)}$
		Precision or Positive Predictive Value $\frac{TP}{(TP+FP)}$	Negative Predictive Value $\frac{TN}{(TN+FN)}$	Accuracy $\frac{TP+TN}{(TP+TN+FN+FP)}$
				F-Measure $2*\frac{Precision*Recall}{Precision+Recall}$

Note(s): TP = number of positive instances correctly identified as positive (the number of students correctly identified as not ‘at risk’); TN = number of negative instances correctly identified as negative (the numbers of students correctly identified as ‘at risk’; FP = number of negative instances identified incorrectly as positive (the number of at-risk students who were incorrectly identified as not ‘at risk’; and FN = number of positive instances incorrectly identified as negative (the number of not-at-risk students incorrectly identified as ‘at risk’

Figure 3.
Confusion metric

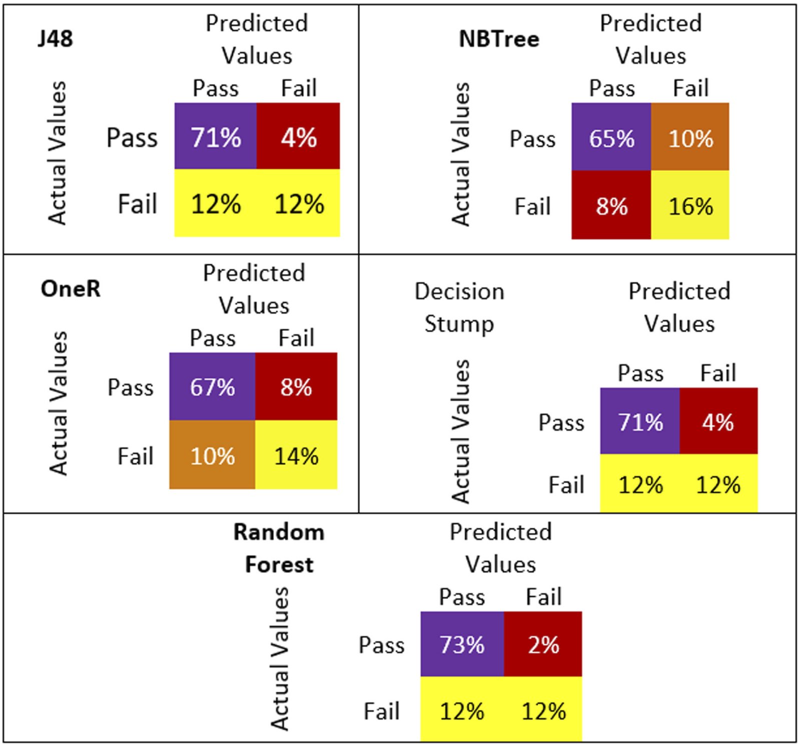
instances, and after ensemble tuning achieved the highest (85.7%) value of all (although the accuracies of other methods are all very similar, ranging 81.6%–83.7%).

A comparison of the performance of the five ML models with and without booster ensemble tuning is presented in Figure 5. ML classification models are more accurate with booster ensemble tuning, and the random forest method again outperforms other classifiers in both cases.

Of the five classifiers, the precision (the ratio of TP to the sum of all positive instances identified by the classifier) and *F*-measure are highest for random forest. Higher precision is preferable because it means fewer instances of FP. The *F*-measure indicates that this classifier has low FP and FN. Ensemble tuning also reduces the prediction error of FP in the “Fail” class using random forest.

Random forest may perform better than other classifiers for a number of reasons. It may improve accuracy because the boosting ensemble method can vote high-ranking instances. It also does not prune trees like other tree-based algorithms. At each tree node, splitting is considered for a random subset of features, resulting in features being split into more and smaller random subsets, increasing the diversity among the forest of trees, leading to its outperformance compared to other decision tree based algorithms. Random forest also uses bagging and generates a forest based on the subset of the model features. The combination of bagging and boosting may reduce overfitting and bias issues, thereby reducing prediction variance.

We introduce an algorithmic method to construct and evaluate ML models to develop an educational decision support system (EDSS) that accurately identifies students at risk of



J48	Predicted Values: Pass	Predicted Values: Fail
Actual Values: Pass	71%	4%
Actual Values: Fail	12%	12%

NBTree	Predicted Values: Pass	Predicted Values: Fail
Actual Values: Pass	65%	10%
Actual Values: Fail	8%	16%

OneR	Predicted Values: Pass	Predicted Values: Fail
Actual Values: Pass	67%	8%
Actual Values: Fail	10%	14%

Decision Stump	Predicted Values: Pass	Predicted Values: Fail
Actual Values: Pass	71%	4%
Actual Values: Fail	12%	12%

Random Forest	Predicted Values: Pass	Predicted Values: Fail
Actual Values: Pass	73%	2%
Actual Values: Fail	12%	12%

Figure 4.
Confusion matrices for the five tree-based classifiers

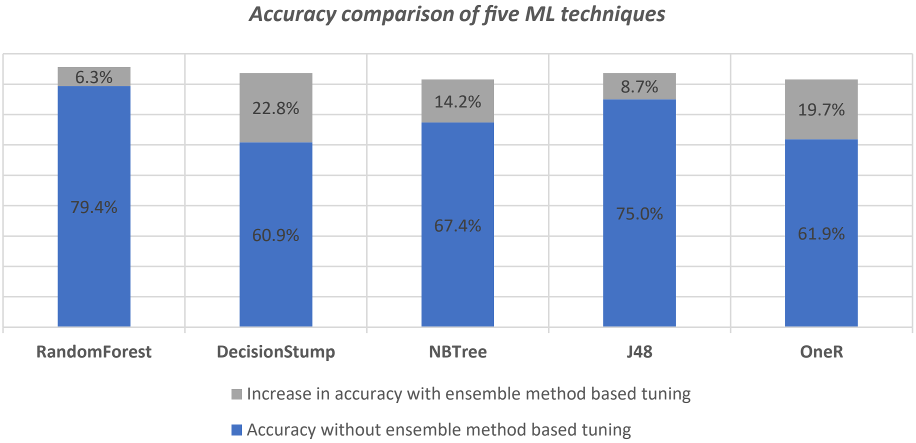


Figure 5.
Comparison of the accuracy of five ML techniques

failure in a timely manner. The dataset that we used comprised the activities and interactions of students with an LMS, and our analysis of it increased the probability of accurately identifying at-risk students early in a semester. Early intervention enables timely implementation of remedial measures to reduce the probability of failure (thereby increasing retention rate). We build the model to analyze the learning behavior of a student which automatically, accurately classifies those students “at risk” of failure.

Our objective is to support educator efforts to improve teaching and learning through the use of an EDSS artifact, instead of using traditional, time-consuming methods that involve a heavy administrative workload. This artifact enables near real-time identification of struggling students’, and the timely implementation of appropriate interventions to enhance their progress. We consider that this timely detection and measurement of at-risk students will contribute to improved progress of some struggling students, increase retention and decrease attrition rates as a consequence and have positive and cascading impacts on the student and institutional reputation and institution financials. Cascading effects could include those on a nation’s economy because it is anticipated that qualified students would be better positioned to repay study debts.

5. Conclusion

We outline a new framework based on ML for improving the academic performance of a student, with appropriate intervention. The proposed ML-based EDSS framework offers better options in terms of the accuracy of classification models. We recognize random forest to be the best of five ML classification algorithms that we appraise at classifying students at risk based on their interaction with LMS. Our automation of this process enables almost real-time identification of at-risk students, which is beneficial from both academic and administrative perspectives. This framework could be set to alert educators to prospective problematic students, triggering the need for support or remedial assistance to facilitate passing.

Future research might be considered using enhanced datasets that incorporate behavioral attributes like interaction with other students, teamwork participation and other student academic attributes to further enhance the model application. Additionally, the dataset could be enhanced by adding new grade levels (other than the binary modes of Pass and Fail), treating it as a multi-class classification problem. Deep learning techniques or classification techniques other than tree-based classifiers and parameter tuning with Weka class-balancer could also be applied, thereby increasing model accuracy.

Note

[1] In this study, we relate student attrition to their readiness in learning and capability development in terms of the talented effort that may contribute to make them succeed in their higher education.

References

1. Mohr JJ, Eiche K, Sedlacek WE. So close, yet so far: predictors of attrition in college seniors. J Coll Stud Dev. 1998; 39(4): 343-54.
2. Wintre M, Bowers C, Gordner N, Lange L. Re-evaluating the university attrition statistic: a longitudinal follow-up study. J Adolesc Res. 2006; 21(2): 111-32.
3. Beer C, Lawson C. The problem of student attrition in higher education: an alternative perspective. J Furth High Educ. 2016; 41(6): 773-84.
4. Imran M, Latif S, Mehmood D, Shah M. Student academic performance prediction using supervised learning techniques. Int J Emerg Technol Learn. 2019; 14(14): 92-104.
5. Zhang Y, Fei Q, Quddus M, Davis C. An examination of the impact of early intervention on learning outcomes of at-risk students. Res High Educ. 2014; 26.
6. Shahiri AM, Husain W, Rashid NZ. A review on predicting student’s performance using data mining techniques. Proced Comput Sci. 2015; 72: 414-22.

7. Conijn R, Snijders C, Kleingeld A, Matzat U. Predicting student performance from LMS data: a comparison of 17 blended courses using Moodle LMS. IEEE Trans Learn Technol. 2017; 10(1): 17-29.
8. Jain A, Solanki S. An efficient approach for multiclass student performance prediction based upon machine learning. In: 2019 International Conference on Communication and Electronics Systems (ICCES) Communication and Electronics Systems (ICCES), Coimbatore, India; 2019: 1457-62.
9. Zeineddine H., Braendle U., Farah A.. Enhancing prediction of student success: automated machine learning approach. Comput Electr Eng; 89: 2021.
10. Tenpipat W, Akkarajitsakul K. Student dropout prediction: a KMUTT case study. 2020 1st International Conference on Big Data Analytics and Practices, IBDAP: Thailand; 2020: 1-5.
11. Purwoningsih T, Santoso HB, Hasibuan ZA. Online learners' behaviors detection using exploratory data analysis and machine learning approach. In: 2019 Fourth International Conference on Informatics and Computing (ICIC), Semarang, Indonesia; 2019: 1-8.
12. Merchan SM, Duarte JA. Analysis of data mining techniques for constructing a predictive model for academic performance. IEEE Lat Am Trans. 2016; 14(6): 2783-8.
13. Shanmugarajeshwari V, Lawrance R. Analysis of students' performance evaluation using classification techniques. International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16); 2016: 1-7.
14. Iatrellis O, Savvas IK, Fitsilis P, Gerogiannis VC. A two-phase machine learning approach for predicting student outcomes. Educ Inf Technol. 2021; 26(1): 69-88.
15. Xu J, Moon KH, Mvd S. A machine learning approach for tracking and predicting student performance in degree programs. IEEE J Sel Top Signal Process. 2017; 11(5): 742-53.
16. Berens J, Schneider K, Gortz S, Oster S, Burghoff J. Early detection of students at risk—predicting student dropouts using administrative student data from German universities and machine learning methods. J Educ Data Mining. 2019; 11(3): 1-41.
17. Kemper L, Vorhoff G, Wigger BU. Predicting student dropout: a machine learning approach. Eur J High Educ. 2020; 10(1): 28-47.
18. Yang F, Li FWB. Study on student performance estimation, student progress analysis and student potential prediction based on data mining. Comput Educ. 2018; 123: 97-108.
19. Li H, Wenbiao D, Songfan Y, Zitao L. Identifying at-risk K-12 students in multimodal online environments: a machine learning approach. in: Rafferty AN, Whitehill J, Cavalli-Sforza V, Romero C (Eds). Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020); 2020. 137-47.
20. Akram A, Fu C, Li Y, Javed MY, Lin R, Jiang Y, Tang Y. Predicting students' academic procrastination in blended learning course using homework submission data. IEEE Access. 2019; 7: 102487-98.
21. Nespereira CG, Elhariri E, El-Bendary N, Vilas AF, Redondo RPD. Machine learning based classification approach for predicting students performance in blended learning. In: Gaber M, Hassanien AE, El-Bendary N, Dey N, editors. The 1st International Conference on Advanced Intelligent System and Informatics (AISI2015), 2015 Nov 28–30. Egypt: Springer; 2016. p. 47-56.
22. Natek S, Zwilling M. Student data mining solution—knowledge management system related to higher education institutions. Expert Syst Appl. 2014; 41: 6400-7.
23. Nghe NT, Janecek P, Haddawy P. A comparative analysis of techniques for predicting academic performance. In: 2007 37th Annual Frontiers In Education Conference—Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007, WI, USA: Milwaukee; 2007: T2G-7-12.
24. Rokach L. Ensemble-based classifiers. Artif Intell Rev. 2010; 33: 1-39.
25. Kotthof L, Thornton C, Hoos HH, Hutter F, Leyton-Brown K. Auto-WEKA 2.0: automatic model selection and hyperparameter optimization in WEKA. J Mach Learn Res. 2017; 18: 1-5.

26. Vahdat M, Oneto L, Anguita D, Funk M, Rauterberg M. A learning analytics approach to correlate the academic achievements of students with interaction data from an educational simulator. In: Conole G, Klobučar T, Rensing C, Konert J, Lavoué É, editors. Design for Teaching and Learning in a Networked World: 10th European Conference on Technology Enhanced Learning, EC-TEL 2015, 2015 Sept 15–18, Toledo, Spain: Springer; 2015: 352-66.

27. Fahd K, Miah SJ, Ahmed K, Venkatraman S, Miao Y. Integrating design science research and design based research frameworks for developing education support systems. Educ Inf Technol. 2021; 26: 4027-48.

Corresponding author
Shah Jahan Miah can be contacted at: shah.miah@newcastle.edu.au