On the Portability and Robustness of Early Student Performance Predictions

Dr. Abdulmalek Al-Gahmi, Weber State University

Dr. Abdulmalek Al-Gahmi is an associate professor at the School of Computing Department of Weber State University. His teaching experience involves courses on object-oriented programming, full-stack web development, computer graphics, algorithms and data structures, and machine learning. He holds a Ph.D. in Computer Science from New Mexico State University.

On the Portability and Robustness of Early Student Performance Predictions

Abstract

The early prediction of student performance has long been a significant area of interest within the educational research community. Numerous studies have sought to identify students who struggle early in their courses, with the goal of providing timely interventions. This early detection of at-risk students is vital for fostering and promoting student success, which is critical to the missions of higher-education institutions and their long-term goals of improving student graduation and retention rates.

This paper builds on the predictive models from a previous paper, focused on making early course-based predictions of student performance based on data collected from a Learning Management System (LMS). It describes these models, augments them with new features, and investigates the portability and robustness of their predictions— aspects not addressed in the previous paper. It frames the early prediction of student performance as a binary classification problem, distinguishing between two student groups: at-risk students who are failing their courses and require early identification for timely assistance, and those who are performing satisfactorily. Additionally, this paper employs Explainable AI (XAI) methods to gain insight into how the new models generate their predictions and to identify the key data features that contribute to these predictions.

This paper relies on data gathered from the Canvas Learning Management System through RESTful and GraphQL APIs. The data pertain to 274 lower-division Computer Science courses, encompassing 2,656 students and 37 instructors. These courses are delivered in various formats (face-to-face, online, and virtual), spanning a three-year period at a public four-year university.

Introduction and background

The field of higher education has witnessed a burgeoning interest in leveraging data and analytics to predict student performance. This interest stems from the institutions' need to understand the factors influencing student success and retention, as well as the desire to offer valuable insights to students and advisors, enabling them to make informed decisions regarding their academic pursuits. The question of whether student performance can be accurately predicted early enough to intervene and provide needed help to struggling students is, particularly, important. For higher education institutions, early detection of at-risk students is essential for planning and providing the appropriate remedial services that students need in a timely manner.

Various approaches to student performance prediction have been explored. Some studies require designing specific randomized experiments [1], [2], [4], [6], while others, like this study, focus on utilizing data gathered by ubiquitous Learning Management Systems (LMSs) based on student activities and interactions with course materials [3], [8], [9]. Additionally, some studies aim to evaluate the efficacy of certain teaching methodologies [4], [5], while others seek to identify problems early in the semester to allow for timely intervention [6], [7], [9]. Like many of these studies, this paper focuses on the early prediction of student performance, leveraging machine learning (ML) algorithms trained on data collected by the LMS.

The use of machine learning and LMS in predicting student performance is not new. For instance, Umer et al. [1] utilized several machine learning (ML) algorithms to predict student outcomes in a course by mining the LMS activity log data. They affirmed the importance of this data in making such predictions but found that it does not necessarily lead to improved predictive accuracy. Similarly, Van Goidsenhoven et al. [2] analyzed activity log data from an LMS to predict student success. They specifically included courses with blended learning environments and discovered that those classes are more challenging to predict student success based on activity streams. Both studies employed a variety of ML algorithms, including random forest and logistic regression. They concluded that while counting activities is helpful in making predictions, it alone is not sufficient.

Shayan et al. [3] studied predicting student performance based on their behavior in an LMS. However, they focused on student performance on formative rather than summative assessments. Conijn et al. [4] investigated predicting student performance by comparing 17 blended courses. They concentrated on examining the portability of predictive models across multiple courses and the timeliness of these predictions. In doing so, they replicated a study by Gašević et al. [5] on the effect of instructional conditions on predicting success, with a larger sample size, using predictors available for all courses. They pointed out that there is a great diversity in the number of variables being used as predictors. Additionally, they noted the inconsistency of findings (and non-robustness) when the same or similar predictors are employed, asserting the necessity to expand the empirical base of the issue of portability, especially as some studies have indicated that prediction accuracy increases over time.

To address the issue of small sample sizes prevalent in previous studies, Gonzalez et al. [6] conducted an analysis of massive LMS log data with the aim of achieving early prediction of course-agnostic student performance. They employed several ML models in a course-agnostic manner to classify students into fail, at-risk, and excellent groups at various intervals (10%, 25%, 33%, and 50%) throughout the course. Data from all courses within a single university over the course of one year were utilized.

Furthermore, Dias et al. [7] introduced DeepLMS: a deep learning predictive model designed to support online learning, particularly in the Covid-19 era. They employed deep learning (DL) techniques to forecast the quality of interaction (QoI) with the LMS using Long Short-Term Memory (LSTM) networks, with RMSE errors as evaluation metrics. Utilizing online learning as a means of mitigating temporal and spatial limitations inherent in traditional courses, they highlighted that a student's QoI serves as a strong efficacy indicator of course design.

Various data sources have been utilized, including fine-grained interaction and activity logs, which are often criticized for their lack of portability and robustness. However, this paper takes a different approach by making predictions based on assessment data (quizzes, exams, assignments, and discussions) extracted from Canvas, a widely used Learning Management System (LMS) in higher education institutions. To the best of our knowledge, none of this data is unique to Canvas or has any specific requirements that other LMSs do not support or possess.

This paper builds upon the work initiated by two recently published papers concerning the early prediction of student performance using only LMS data [8], [9]. While it summarizes the models

of these two papers and augments them with new data features, its primary focus lies on investigating the portability of these models and the robustness of their predictions, aspects which have not been previously explored. But what do portability and robustness mean, and why are they important?

In the context of this paper, portability refers to the adaptability and effectiveness of a predictive model when applied to different educational settings or teaching styles. A portable model should maintain its predictive accuracy and generalizability when trained on different datasets pertaining to various educational settings, such as courses with different modalities, different semesters, or taught by different instructors. Robustness, on the other hand, refers to the ability of a trained predictive model to sustain its performance in the face of variations, uncertainties, or changes in the data distribution or input conditions. A robust model should not be overly sensitive to minor changes in the input data, such as variations in data quality or changes in the student population. It should provide reliable predictions under a range of conditions.

There are several reasons why portability and robustness are crucial. First, institutions and educational environments can vary significantly in terms of teaching methods, curriculum structure, student demographics, and other factors. Portability ensures that a model performing well in one setting can be applied to another, while robustness guarantees that a model trained in one setting remains applicable in others. Secondly, educational environments are dynamic, and conditions can change over time. Robust, portable models are less affected by changes in data distribution, policy adjustments, or shifts in teaching approaches.

The rest of this paper is organized as follows. The next section discusses the approach taken by this paper regarding data acquisition, cleanup, and preprocessing, as well as feature engineering and models. The subsequent section explores the preprocessed data and its diversity. Following that, the paper presents and discusses the results of all the experimentation conducted. Finally, the last section delves into future work and offers concluding remarks.

Approach

The early prediction of student performance is framed as a binary classification problem with two distinct classes: POSITIVE (coded as 1) for at-risk students and NEGATIVE (coded as 0) for all others. This classification is based on a threshold of C- (cumulative score of <74%) or below. Predictions are generated by models trained on data, utilizing the effective models identified in previous studies [8], [9].

All data utilized in this study were extracted from the Canvas Learning Management System, collected via RESTful and GraphQL APIs from 274 lower-division Computer Science (CS) courses. These courses encompass 2,656 students and 37 instructors and are offered in various formats, including face-to-face, online, and virtual classes. The data span a three-year period at a public four-year university. These courses exhibit diversity in terms of topics, sizes, levels (1st year and 2nd year), modality (face-to-face, virtual, and online), semester (spring, summer, fall), and instructors. The objective is to leverage these variations to evaluate the portability and robustness of the predictive models.

To commence, determining the timing of feature computation for each student enrolled in a course presents a challenge in a dataset as diverse as this. Selecting a single point in time, such as February 9, 2022, proves impractical since some courses have concluded while others have yet to begin. Instead, time points are made relative to the course's timespan, utilizing a fixed-length timeline from 0 to 100. This approach ensures that each time point represents a percentage of the course completion. A concise overview of how this data is acquired, cleaned, and prepared is provided in the following subsections.

To address the question of portability and robustness of these models, this paper undertakes the following steps:

- It delves into the data, illustrating how two different students (one at-risk and one performing satisfactorily) are represented using the aforementioned time-based representation. Additionally, it explores the diversity of this data across courses and students at various levels, modalities, semesters, and taught by multiple instructors. Furthermore, it discusses the challenges associated with working with this inherently unbalanced data.
- It enhances the existing models with new features related to specific course events, such as missing assignments, making late submissions, or failing individual assignments.
- It conducts numerous experiments, training a variety of machine learning models using the augmented data to evaluate their portability and the robustness of their predictions.

Data Acquisition

The data analyzed in this paper comprises 10 required lower-division CS courses taught over a span of three years, from Spring 2019 to Summer 2022, catering to students pursuing their associate CS degrees within the same timeframe. Across these courses, there were a total of 274 sections, averaging 27.6 sections per course. Being mandatory CS courses, they typically accommodate more students and are offered in various modalities compared to other courses. Many of these courses are offered multiple times in different formats by various instructors within the same semester.

Primarily relying on the Learning Management System (LMS) as the main platform for instruction, these courses utilize the LMS for posting learning materials, facilitating discussions, and collecting assignments and other graded activities. The LMS meticulously records all activities and events that occur within its interface. In addition to basic student information, it contains data pertaining to assignments, quizzes, and other graded activities, including submission attempts, scores, and due dates, among other details. Moreover, it maintains activity logs documenting student interactions with resources such as pages, modules, or assignments, including details about what was accessed, when, and how frequently. This paper specifically focuses on the LMS data related to assignments and other graded activities, leveraging this rich dataset to explore early prediction of student performance.

Data Cleanup and Preparation

The data underwent two preparatory steps: anonymization and normalization. In the anonymization step, randomly assigned IDs were used instead of identifying names for course sections, instructors, students, assignments, and assignment groups.

The normalization step ensured that the possible total score at the end of each course added up to 100%. This was crucial to prevent discrepancies between scores on different assessments. For instance, a score of 90% on a quiz worth 5% of the final score should not equate to the same as a score of 90% on an exam worth 30% of the final score. Normalizing these scores was a complex task, as courses were set up differently and required different normalization approaches. All calculations were rigorously verified by comparing the cumulative scores at the end of each course to the actual final scores obtained from the LMS.

Courses without any student activity in the LMS were removed from the dataset. Additionally, students whose cumulative normalized scores did not align with their final scores were excluded from the final dataset. The resulting dataset consists of time series sequences indexed by student IDs, course IDs, and timestamps. It includes columns for normalized and possible scores, as well as cumulative normalized and possible scores.

Finally, to standardize the representation of each student in every course, a fixed length of 100 was applied to each time series sequence. This choice allows us to think of each data point as representing the student's status at a specific percentage point of the course. For longer time series, where two or more data points needed to be combined, a new time point was created by summing the normalized and possible scores. This process maintains the order of events (except for combined time points) and preserves the relative distance between data points.

Feature Engineering

In addition to compressing and fixing the length of the time series sequences, five additional quantities were calculated at a given point in time, t:

- Missed opportunity: This represents the amount of coursework that the student has missed up to that point in time and is calculated as:

$$\text{missed\_opportunity}_t = \text{possible\_score}_t - \text{actual\_score}_t$$

- Relative achievement: This indicates how much of what is possible for a student to achieve has been accomplished and is calculated as:

$$\text{relative\_achievement}_t = (\text{actual\_score}_t / \text{possible\_score}_t) \times 100$$

- Number of missed assessments: A missed assessment is defined as one for which the student achieved a zero score out of a non-zero possible score they could have achieved.

- Number of late assessments: A late assessment is defined as one that does not align timewise with its corresponding possible score, indicating it was not submitted on time.

- Number of failed assessments: This represents the count of assessments that the student submitted but failed to achieve a passing score.

While only the first two features were utilized by the models in the previously cited work, this paper introduces the last three features to enhance model robustness and smooth out learning.

Predictive Models

The previous papers identified four machine learning models that performed well [8][9]. Table 1 lists these models along with the hyperparameters they were trained with in this paper.

Table 1: Models used and their hyper-parameters values

Algorithm	**Parameters**
Decision Tree	max_depth=3
Random Forest	max_depth=3
Logistic Regression	C=1e5
Multilayer Perceptron	max_iter=1000

These models, implemented by the SciKit Learn library [10], are trained at various course percentages using data containing the aforementioned five features.

Data Exploration

To gain insight into the preprocessed dataset, Figure 1 (top) shows the cumulative normalized and possible scores of two randomly selected at-risk and passing students from the dataset. The progression of a student's time in a course is depicted as an upward stair-like pattern. The width of the horizontal steps is determined by the number and distribution of graded activities throughout the course, while the height reflects the weights of these assignments and activities.

A student's struggle in a course can be visualized by the difference between their cumulative normalized and possible score curves. This difference tends to increase over time, particularly for at-risk students, indicating a steeper decline in performance. Additionally, this data allows for comparisons between the progress of an average at-risk student and an average passing student. Figure 1 (bottom) displays such progressions side by side, with averaging resulting in smoother, almost linear curves, while maintaining similar gaps between actual and possible scores.

The dataset encompasses journeys similar to these for each of the 2,656 students across all sections of the ten CS courses outlined in Table 2. This table highlights the diversity of the dataset. The first three courses are at the first-year level, while the remaining are second-year level courses. All courses are offered in all three semesters (Spring, Summer, and Fall). Notably, the initial three courses attract a larger number of students, including many non-CS students from engineering and other majors, potentially leading to a different student distribution compared to second-level courses.

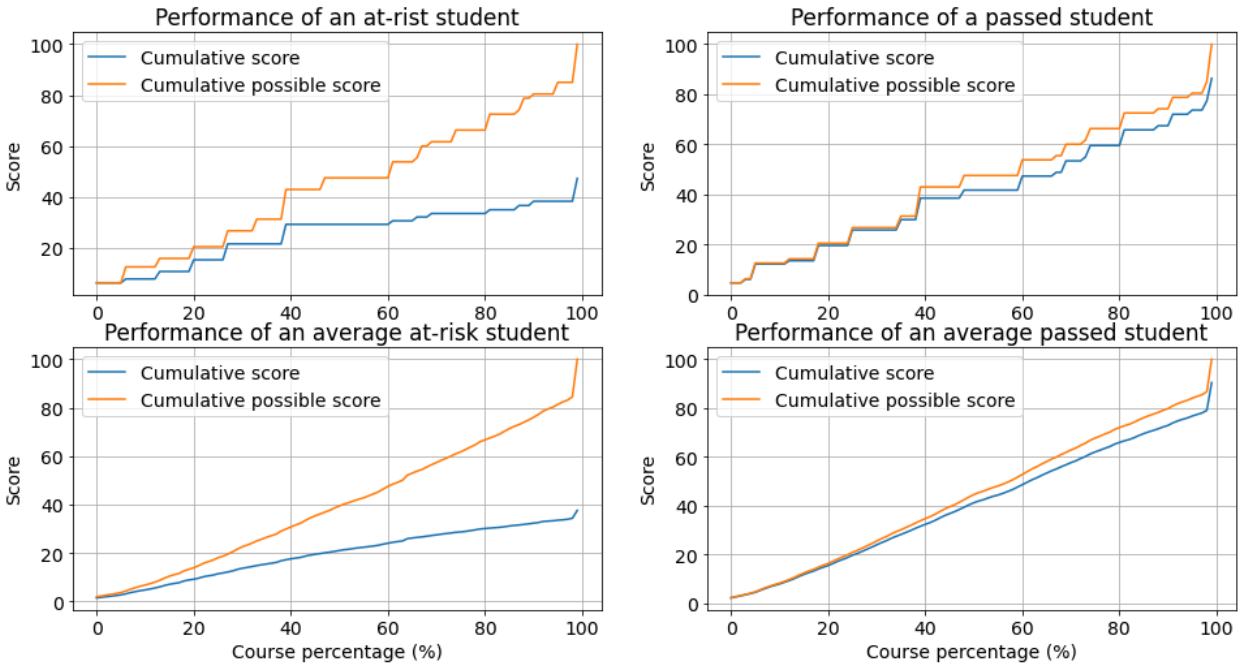

Figure 1: The progression of an actual and an average at-risk and passing students.

Lastly, it is crucial to acknowledge that datasets like this are inherently unbalanced, with a notably larger number of passing students compared to failing ones. This disparity needs to be carefully considered during the training and evaluation of predictive models, as it can significantly impact the interpretation and utility of the accuracy metrics.

Table 2: Courses' sections, modalities, instructors and students.

#	**Course**	**Modality**	**# of Sections**	**# of Instructors**	**# of Students**
1	Computing Foundations	F2F, ONL, VTL	47	10	1210
2	Programming I	F2F, ONL	35	13	773
3	Object-Oriented Programming	F2F, ONL, VTL	20	10	607
4	Computational Structures	F2F, ONL, VTL	26	7	562
5	Client Side Web Development	F2F, ONL, VTL	25	7	565
6	Data Structures & Algorithms	F2F, ONL, VTL	19	5	532
7	Software Engineering I	F2F, ONL, VTL	23	5	497
8	Database Design & SQL	F2F, ONL	40	8	843

9	Network Fundamentals	F2F, ONL	20	3	509
10	Computer Architecture	F2F, ONL	21	5	534

Results

As stated previously, the primary focus of this paper is to assess the portability of the utilized machine learning models and the robustness of their early student performance predictions. Figure 2 (left) displays the cross-validation accuracies of all four models at various time points (percentages) throughout the courses, utilizing the entire dataset.

It is evident from this figure that all four models exhibit similar performance trends. For instance, at 40% of the course duration, all models predict whether the student will be at-risk or not at the end of the course with approximately 87% accuracy. The red line at the bottom serves as a reminder of the unbalanced nature of the dataset. A dummy model that consistently predicts the negative class (Passing) would achieve an accuracy of 78%. As expected, the trained models consistently outperform this baseline, and their performance tends to improve over time.

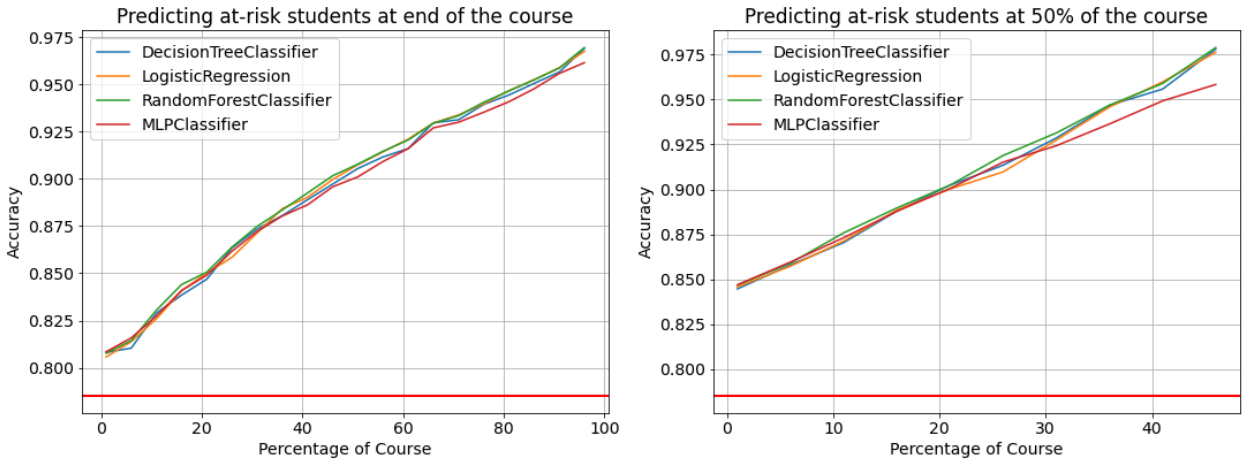

Figure 2: Model performance at the end of the course (left) and 50% of the course (right).

Figure 2 (right) highlights another aspect of these models. Predictions do not necessarily need to be based on a distant future point, such as the end of the course. It is possible to make predictions about student performance at nearer future points within the course. The figure shows the performance of the same models in predicting which students will be at-risk and which will pass at the midpoint of the course. Interestingly, shortening the prediction horizon results in higher performance.

Additionally, by utilizing SHAP values [11], it is possible to examine how the aforementioned models generate their predictions. Figure 3 shows the important features of the RandomForest model, indicating that relative achievement is the most influential feature, followed by missed opportunity and the number of missing submissions.

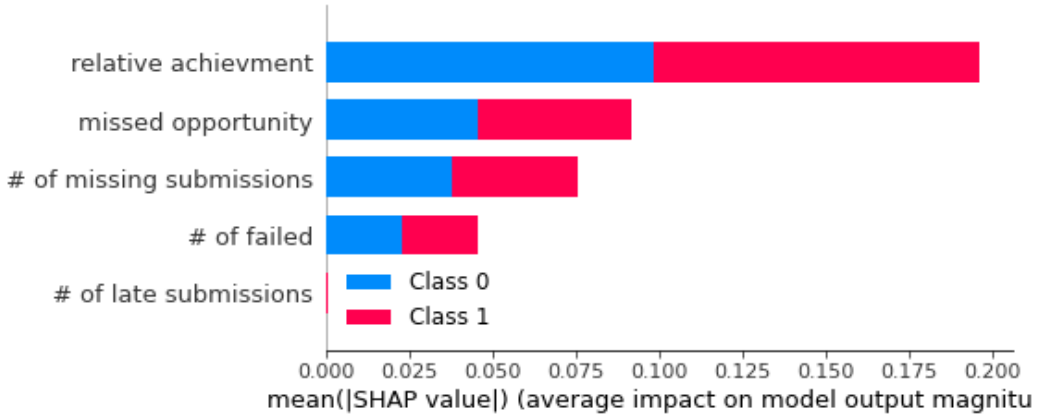

Figure 3: Random Forest's important features at 30% of the course.

Model Potability

To assess the portability of the models, multiple experiments were conducted by training models using different variations of the dataset. However, considering the diversity of the dataset, as indicated in Table 2, training models on all combinations of levels, semesters, modalities, courses, and instructors is impractical.

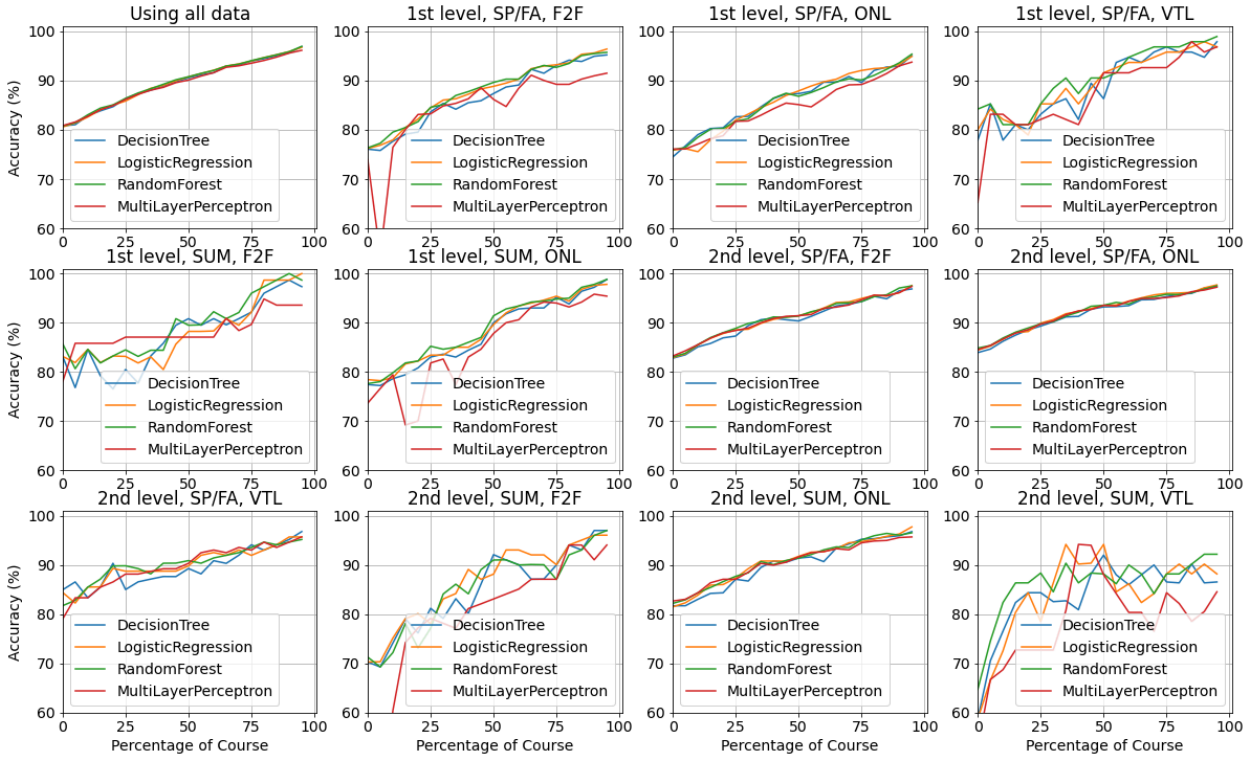

Figure 4: Models trained on different variations of the data.

In this paper, numerous experiments involving various combinations were conducted, and this subsection presents the results of one such experiment. Here, courses were divided into levels,

and semesters were categorized as either regular Spring/Fall (SP/FA) or summer (SUM) semesters. The dataset was then split into eleven parts, each representing a combination of level, semester, and modality. Subsequently, instances of all four models were trained and cross-validated on these sub-datasets at various percentages of the course. Figure 4 shows the performance of these models compared to the ones trained on the entire dataset (top - left).

.As can be seen, the models demonstrate similar performance across all these variations, suggesting their portability to different educational settings. Figure 5 depicts the performance of the aforementioned models when trained on these eleven different sub-datasets and evaluated at 30% of the course. Once again, their performances exhibit consistency.

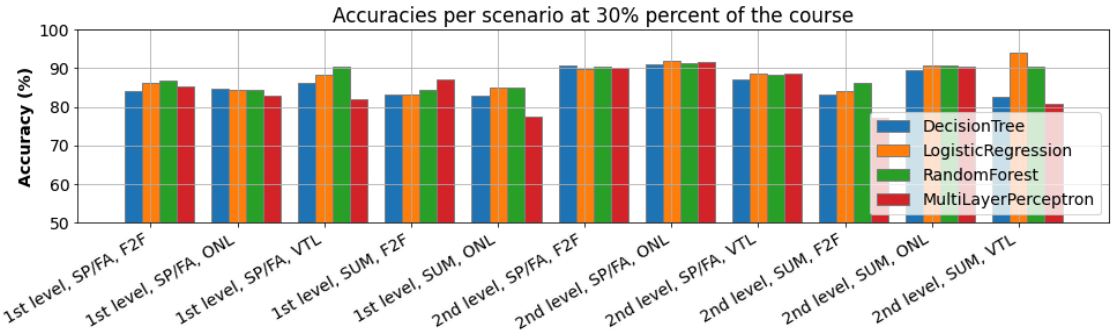

Figure 5: Model performance per data variation at the 30% point of the courses.

To delve deeper into this analysis, paired cross-validated t-tests, utilizing the 5x2cv t-tests method [12], were conducted to compare these models across the datasets. The obtained p-values from these tests were all > 0.05, indicating that the null hypothesis of these models performing similarly across the sub-datasets cannot be rejected. Thus, there is no significant difference in their performances.

Other experimentations with various data variations, which are not reported here for brevity, yielded similar outcomes. In summary, the four models (DecisionTree, LogisticRegression, RandomForest, and MultilayerPerceptron) exhibited satisfactory performance when trained on diverse datasets. This indicates that these models are portable and can be effectively applied to the different educational settings encoded in this dataset.

Prediction Robustness

Similarly, numerous experiments were conducted to assess the robustness of the predictions made by the four models. Unlike the portability experiments, here we first train the four models on a specific subset of the dataset. Then, without retraining them, we evaluate the trained models on other variations of the dataset. This approach allows us to determine whether these models generalize well to different unseen data and how they perform when student distribution changes.

One such experiment involves dividing the dataset into 36 subsets based on levels, years, semesters, and modality. The four models were trained on the first data subset (1st level, 2019, Spring/Fall semester, and face-to-face). These trained models were then evaluated on the remaining 35 subsets.

Figure 6: Models trained on the first of 36 subsets of the data and evaluated on the rest.

Figure 6 shows the performance curves of these models across all 36 data subsets, while Figure 7 displays the performance of the same models evaluated on all 36 data subsets at the 30% point of the courses.

As you can see, the models exhibited strong performance across most of these data variations. However, there is more variability in these performances compared to what was observed under the model portability experiments. This suggests that while these models generally perform well, their predictions may not be equally robust across all settings.

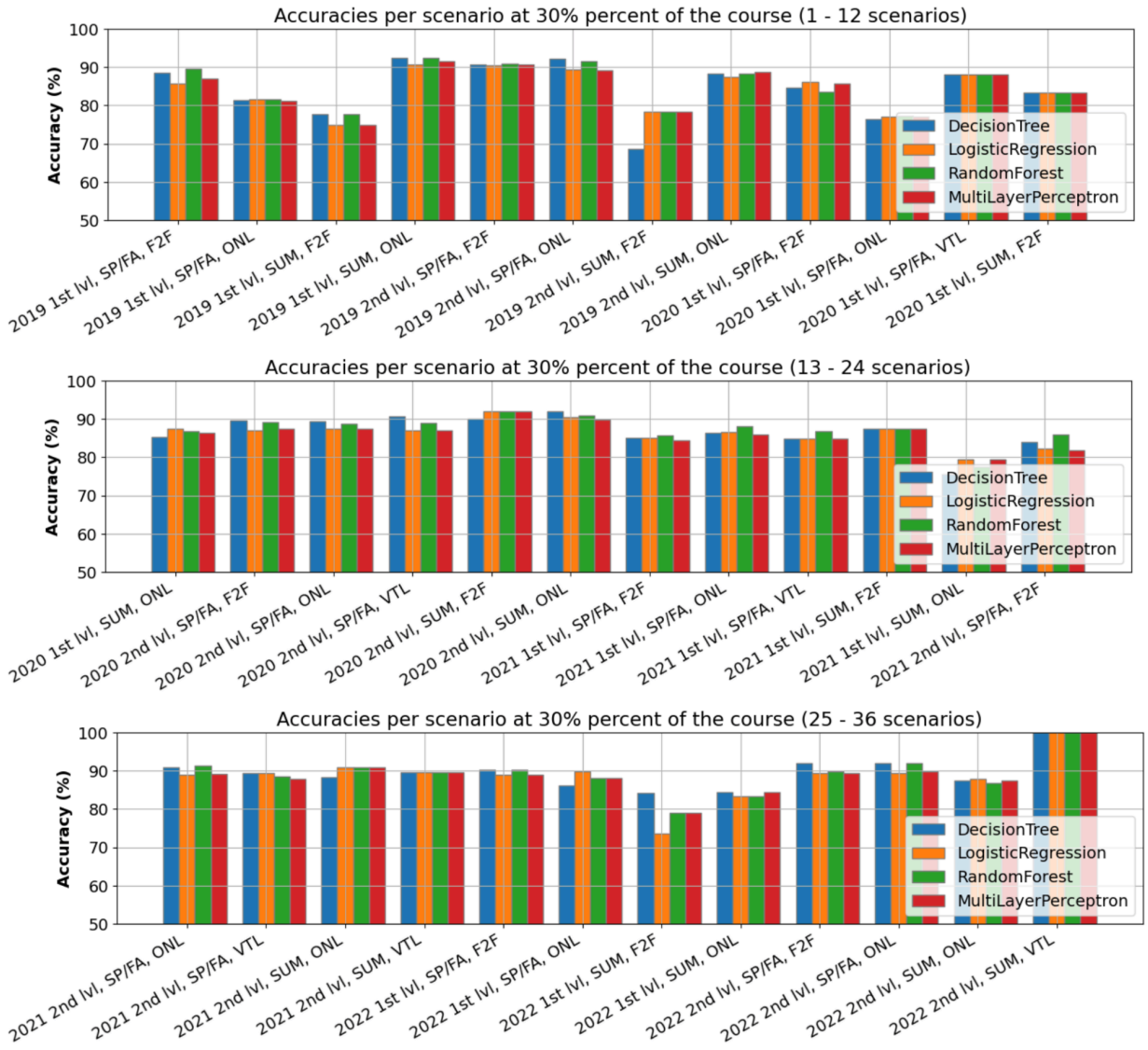

Figure 7: Models' performance per data variation at the 30% point of the courses.

Concluding Remarks and Future Work

As shown in the preceding sections, the early prediction of at-risk students can be achieved with good accuracy. The consistency of the results from different models underscores the robustness of these predictions. The results also show that these models are portable to various settings,

encompassing courses with different modalities taught by diverse instructors. While predictions may exhibit some variation, they remain reasonably robust.

These findings also imply that students' progressions throughout courses exhibit similarities regardless of the specific course or instructor. This suggests the presence of overarching patterns that recur across multiple courses, sections, instructors, and modalities. However, the exploration of localized patterns unique to individual students remains an open and intriguing question.

The models presented in this paper require further refinement before they can be deployed for production use. One potential avenue for improvement is to augment the dataset with relevant student data from other sources besides the LMS. As discussed in the background section, student performance in courses is influenced by various factors, and incorporating these factors could enhance the performance of the models presented here. Additionally, further analysis is warranted to study student trajectories across multiple courses. The current dataset allows for tracking students as they progress through their associate degree, making it possible to consider the course as seasonality of these multi-course time series sequences. Furthermore, it is worth investigating whether integrating activity counts into the models described in this paper can enhance their performance, given that previous research suggests activity counts alone are insufficient predictors of student performance [2].

While the results of this paper are based on lower-division CS courses, it is reasonable to assume they may apply to other CS and non-CS courses. However, it remains an open question whether and how these findings will generalize to other non-CS courses.

In summary, this paper illustrated how time series sequences of graded activities can offer insights into student progression through courses. It evaluated various machine learning models for the task of making early predictions of student performance and assessed their portability and the robustness of their predictions. Having highly accurate models that are reasonably portable and produce robust predictions is crucial for higher education institutions to provide timely support to struggling students, thereby improving learning outcomes and student retention.

Bibliography

[1] R. Umer, A. Mathrani, T. Susnjak and S. Lim, "Mining Activity Log Data to Predict Student's Outcome in a Course," in Proceedings of the 2019 International Conference on Big Data and Education, New York, NY, USA, 2019.

[2] S. V. Goidsenhoven, D. Bogdanova, G. Deeva, S. v. Broucke, J. D. Weerdt and M. Snoeck, Predicting Student Success in a Blended Learning Environment, New York, NY, USA: Association for Computing Machinery, 2020.

[3] P. Shayan and M. v. Zaanen, "Predicting Student Performance from Their Behavior in Learning Management Systems," International Journal of Information and Education Technology, vol. 9, no. 01, pp. 337-341, 2019.

[4] R. Conijn, C. Snijders, A. Kleingeld and U. Matzat, "Predicting Student Performance from LMS Data: A Comparison of 17 Blended Courses Using Moodle LMS," IEEE Transactions on Learning Technologies, vol. 10, no. 01, pp. 17-29, 2017.

[5] D. Gašević, S. Dawson, T. Rogers and D. Gasevic, "Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success," The Internet and Higher Education, vol. 28, pp. 68-84, 2016.

[6] M. Riestra-Gonza´lez, M. d. P. Paule-Ruíz and F. Ortin, "Massive LMS log data analysis for the early prediction of course-agnostic student performance," Comput. Educ., vol. 163, pp. 104-108, 2021.

[7] S. B. Dias, S. J. Hadjileontiadou, J. Diniz and L. J. Hadjileontiadis, "DeepLMS: a deep learning predictive model for supporting online learning in the Covid-19 era," Scientific Reports, vol. 10, no. 1, p. 19888, 2020.

[8] A. Al-Gahmi, K. D. Feuz, Y. Zhang, "On Time-based Exploration of LMS Data and Prediction of Student Performance,". 2022 ASEE Annual Conference & Exposition, Minneapolis, MN, June 2022, 10.18260/1-2--40852

[9] A. Al-Gahmi, K. D. Feuz, & Y. Zhang, "On Time-based Exploration of Student Performance Prediction", 2023 ASEE Annual Conference & Exposition, Baltimore , Maryland, June 2023, 10.18260/1-2--43772

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg and J. Vanderplas , "Scikit-learn: Machine Learning in Python," The Journal of Machine Learning Research, vol. 12, p. 2825–2830, 2011.

[11] S. Lundberg, S. Lee,"A Unified Approach to Interpreting Model Predictions". Advances in Neural Information Processing Systems 30, NIPS 2017.

[12] T. Dietterich, "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms". Neural Comput 10, p. 1895–1923, 1998.