ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE

Paper ID #36962

On Time-based Exploration of LMS Data and Prediction of Student Performance

Abdulmalek Saleh Al-gahmi (Assistant Professor)

Dr. Abdulmalek Al-Gahmi is an assistant 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, M.S. in Computer Science, M.A. in Extension Education, and B.S. in Electrical Engineering.

Kyle Feuz

Kyle Feuz is an Associate Professor at Weber State University in the School of Computing. He earned his Ph.D from Washington State University under the guidance of Dr. Diane Cook in 2014. He also received his B.S and M.S in Computer Science from Utah State University in 2011.

Yong Zhang

Dr. Yong Zhang is an associate professor in Computer Science at Weber State University. He received the B.E. degree and M.E. degree in Electrical Engineering from Harbin Institute of Technology, China, and the Ph.D degree in Electrical Engineering from West Virginia University, Morgantown, USA. His research interests include digital image and video processing, bioinformatics, and computer vision.

© American Society for Engineering Education, 2022
Powered by www.slayte.com

On Time-based Exploration of LMS Data and Prediction of Student Performance

Abstract

Learning Management Systems (LMS) gather extensive amounts of data about students' progression through courses. Such data is then made available via APIs for data exploration and utilization. A body of research has investigated using such data to predict student performance based on data collected earlier in a course. The driving question in such efforts has been whether student performance can be accurately predicted early enough to intervene and provide needed help. Two main issues have been pointed out in these attempts: the portability and robustness of these predictions.

In this work-in-progress study, we introduce a new approach to exploring LMS data. Such an approach looks at the data as a set of time series each representing the progress of a student within a course. This study explores how students advance through courses over time and the variability of student performance between any two time points. It utilizes the LMS data (Canvas in this case) obtained from multiple Computer Science courses taught by different instructors in different formats (online and face-to-face) over three years in a public four-year university. Ways for summarizing and visualizing such data are discussed, and useful predictor features are extracted and used to build and evaluate predictive models of student performance at any time point. The study explores questions such as: How does the progress of passed and failed students differ in these courses? How early can student performance be accurately predicted? Can data collected from one course be used to predict the performance of students in another course by the same or a different instructor? Are student journeys through courses unique, or are there patterns that transcend students and courses?

Introduction

Early detection of at-risk students is vital to fostering and promoting student success, which is critical to the mission of any higher-education institution. It allows for planning and providing the appropriate remedial services that students need in a timely manner. It requires the ability to predict student performance several times throughout courses. Many predictive models have been proposed and used to varying degrees of success to make such predictions. Some of these models are at the exam level, some at course level and some at the degree level. These models require the use of datasets that typically come from multiple sources such as student information systems (SIS's) and pre-college information, to name a few.

Learning Management Systems (LMS's) which are widely used by higher-education institutions are an important source of student-related data. They provide a convenient and important way to deliver learning materials to students. They also are the places where most of the course discussions, student-instructor interactions, and assessments take place. Conveniently they keep an extensive record of all such activities and make that data available using dedicated API services. A body of research has investigated using such data to predict student performance at the course level. One question that is frequently posed in these studies has been whether student performance can be accurately predicted early enough to intervene and provide needed help to struggling students.

In this work-in-progress study, we explore the data extracted from an LMS for three Computer Science (CS) courses taught by three different instructors at a public four-year university. These courses use the LMS to deliver learning materials, assess students, and give them feedback. Each course has multiple sections taught over a couple of semesters in multiple modalities (face-to-face, hybrid, virtual, and/or online). The goal is to explore such data, gain an insight into how students advance through courses, and use that insight to propose and evaluate predictive models that can accurately predict at-risk students at the course level in a timely manner. We are also interested in how such insight could be used to implement early intervention and improve student retention.

The first step to answering this study's questions requires creating and curating an appropriate dataset (or datasets). All the data used in this study is obtained via API calls from Canvas: the LMS that our university uses. To our knowledge, none of this data is unique to Canvas or has any specific requirement that other LMS's do not support or have.

This paper reports on the portion of this study that focuses on exploring and utilizing the LMS data related to both formative and summative assessments and participation in graded activities. Here we propose and utilize a new approach to looking at this data progressively. More specifically, each course is looked at as a collection of time series, each of which represents the progress of a single student in that course. We explore such data and devise certain predictor features that might be useful to the task of predicting student performance using supervised machine learning (ML) models. In this paper, such prediction is framed as a binary classification problem of whether a student will pass or fail at the end of the course. Students predicted to fail a course are considered at-risk, and the timeliness of making such predictions with adequate accuracy is critical to being able to provide these students the help they need.

The rest of this paper is organized as follows. The next section briefly reviews the recent body of research that utilizes LMS data to predict student performance. The section after that discusses the methodology/approach used in this study. We then report and discuss the performance of the devised predictive models. A discussion of future work is presented before this paper concludes.

Background

Developing models for predicting student performance in a course has long been a topic of interest. Some of that interest is driven by a desire to evaluate the efficacy of certain teaching methodology[4], [5], while others seek to catch problems early enough in the semester to still have time to intervene [6], [7]. Some of these studies require designing certain randomized experiments [1], [2], [4], [6]. Yet there are studies similar to ours that focus on utilizing the data that the ubiquitous LMS's gather based on student activities and interactions with the course materials. In this section, we review some of that work.

Umer et al [2] use several machine learning (ML) algorithms to predict student outcomes in a course by mining the LMS activity log data. They confirm the importance of LMS data in making such predictions but find out that having LMS data does not necessarily lead to improved predictive accuracy. Similarly, Van Goidsenhoven et al [3] also analyze activity log data from the LMS to predict student success. They specifically include courses with blended learning environments and discover that those classes are harder to predict student success based upon LMS activity streams. Both studies use a variety of ML algorithms including random forest and logistic regression. They conclude that LMS data is helpful in making predictions about student success but counting activities is not good enough.

Similarly, Shayan et al [1] studies predicting student performance based on their behavior in an LMS. However, they focus on student performance on formative rather than summative assessments.

Conijn et al [4] studies predicting student performance by comparing 17 blended courses using the Moodle LMS. They focused on studying the portability of predictive models across multiple courses and the timeliness of these predictions. In doing so, they replicate a study by Gašević et al [5] on the effect of instructional conditions on predicting success with a bigger sample size using predictors available for all courses. They point out that there is a great diversity in the number of variables being used as predictors. They also point out the inconsistency of findings (and non-robustness) when the same or similar predictors are used and claim that there is a need 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 size that previous studies suffer from, Gonzalez et al [7] analyzes massive LMS log data for the purpose of achieving early prediction of course-agnostic student performance. They use several ML models in a course-agnostic way to classify students into fail, at-risk, and excellent groups at 10%, 25%, 33%, and 50% of the course. All courses for one year in a single university are used. Furthermore, Dias et al [6] proposes DeepLMS: a deep learning predictive model for supporting online learning, especially in the Covid-19 era. They use deep learning (DL) techniques to forecast the quality of interaction (QoI) with LMS using LSTM networks with RMSE errors. They use online learning as a way of reducing temporal and spatial problems found in traditional courses. They indicate that the QoI of a student is a strong efficacy indicator of the course design. They use three datasets (DB1-DB3) from three different countries (Portugal, UAE, Greece) with DB1 being pre-Covid and DB2 & DB3 post it.

In summary, LMS data has been used to predict student performance. The driving question in many of these studies has been whether student performance can be accurately predicted early enough to intervene and provide needed help. Most of the previous studies make use of fine-grained interaction and activity logs which suffers from a lack of portability and robustness, especially in the face-to-face or blended learning environments. In this study, We seek to make predictions based on data that is readily available regardless of the class modality. The only requirement is that students submit their work using an LMS. We also propose a new approach to look at each student's performance as a time-series and consider new features that can be derived to improve predictions when using this model.

Approach

This study looks at a student's journey through a course as a multi-step process with ups and downs. Such a process is better captured as a time series rather than as a single data point. We believe this leads to a finer-grained understanding of this journey and better utilization of the readily available LMS data. In this paper, we are interested in the time-based exploration of this data and the application of traditional supervised machine learning models to it.

The first step is to identify the courses suited to this study. The following three courses were chosen:

- Object-Oriented Programming: A required lower-division course
- Introduction to Data structures and Algorithms: This is another required lower-division course.
- Formal Languages and Algorithms for Computing: A required upper-division course.

These courses are selected for many reasons. First, as required CS courses, they tend to have more students. They are also offered in more modalities than other courses. Secondly, our internal data shows that we lose many students in the transition from the first course to the second. This makes these courses appropriate for a study like this, where the goal is to predict student performance and identify at-risk students early enough to intervene in a way that improves retention and reduces dropout rate. Thirdly, the quantity and quality of the data obtained from the LMS about a course depend on how much the LMS is used in that course. Courses that do not use the LMS much will not have enough data to drive a study like this; at least not in the granularity, we would like to see. The courses above use the LMS as the primary place of instruction where learning materials are posted, discussions ensue, and assignments and other graded activities are submitted.

The next step is to select which data to obtain from the LMS. The LMS keeps an extensive record of all the activities and events that take place in it. In addition to the basic information about students, it has data about assignments, quizzes, and other graded activities including submission attempts, scores, and due dates to name a few. There are also activity logs that it keeps about what, when, and how many times a student accesses a certain resource like a page, a module, or an assignment. This paper focuses on the LMS data related to assignments and other graded activities. This data consists of three sets pulled separately and then joined.

- Data about the course: name, begin and end dates.
- Data about the assignments and other graded activities: names, groups, weights, rules, and total possible points.
- Data about student interactions with these assignments and graded activities: submission attempts, lateness, and grades.

Two types of APIs were used to obtain these sets of data: RESTful and GraphQL. This is due to the nature of our LMS in which the GraphQL APIs, although more convenient, are still a work in progress. The data is obtained in JSON format and has to be converted to a tabular format that is more fitting for data analysis.

Two preparatory steps were performed on this data. The first step was to anonymize both the courses from which this data came and their students. The second step was to standardize/normalize all the graded activity scores for all students in all courses. This is to make sure that for any student, the possible total score at the end of the course will add up to 100%. This is also necessary to make sure that a score of 90% on a quiz that is worth 5% of the final score, for example, is not the same as a score of 90% on an exam worth 30% of the final score.

The next step is to make sense of the collected data. This study thinks of every student within a course as a time series over the span of that course's semester. Assignment submissions, quiz attempts, and discussion posts, all, become events with time points in these time series. When one time point corresponds to multiple activities, these activities are grouped into one event whose score is the sum of these activities' scores. The granularity of these time series (daily, weekly or bi-weekly) could vary depending on the course structure. This study uses days as time points.

Treating students as time series makes it easier to handle course sections that vary in length or are structured differently. This is important because even sections of the same course by the same instructor change over time; new assessments and other activities are added and/or removed. The resulting time-based dataset allows for exploring the data to gain insights into how students advance through courses. The goal of this exploration is to find patterns and identify important features that can be used later to predict student performance using traditional supervised machine learning models.

Student performance is typically measured by the final score at the end of the course, and this score is the result of accumulating the student's scores from all the assignments and graded activities in the course. It makes sense then to use cumulative scores in these time series. In other words, the score of a student at any given time point is the sum of all of her scores from the beginning of the course up to that time point. The possible score can also be added cumulatively. This is the score of a hypothetical student getting 100% on every assignment and graded activity at any given time point. Using the possible scores, we can compare the actual performance of a student to what is possibly achievable. Both actual scores and possible scores allow us to visualize how students advance through courses using stepwise upward stair-like curves such as the ones shown in Figure 1. Here a struggling student is defined as one with a less than 74% final score at the end of the course. A passing student is one with a 74% or more final score. Furthermore, the horizontal steps are controlled by how many assignments/graded activities are in and how they are distributed throughout the course. The vertical steps are controlled by the weights of these assignments and activities.

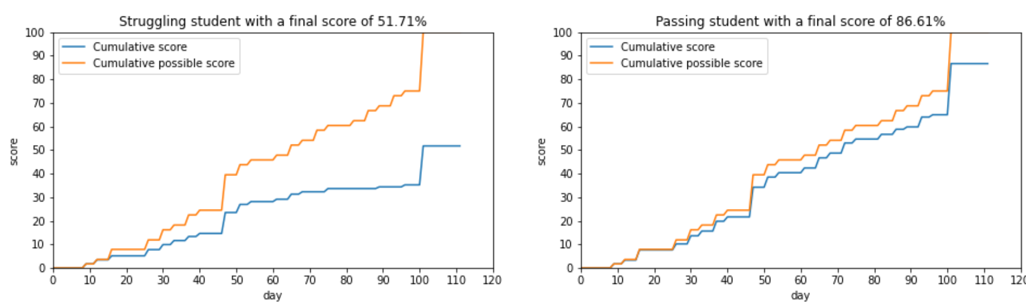

Figure 1: The progress of two students through a course

The cumulative actual and possible scores can be used to derive a few features that might be useful in predicting student performance at any given time point. First, we can draw a line from point (0,0) at the bottom left corner to any other point on the score curve. This line has the property that at the end of the course, its slope times 100 matches the actual final score of the student. This is because the slope of this line is standardized such that a 100% score at the end of the course corresponds to a slope of 1. We call this the *standardized slope* and its value at point (x, y) is calculated using the formula:

$$standardized\ slope(x, y) = \frac{y}{x} \times \frac{x_{max}}{y_{max}}$$

where x_{max} is the maximum number of days in a course time span and y_{max} is the maximum score, which is 100. Since each course section is different in its time length, standardizing the slope allows us to combine data from multiple sections into the same dataset. We call this line the *score line* and its slope changes from one point to the next.

Furthermore, two additional features are calculated from the above actual and possible scores: missed opportunity and relative achievement. The missed opportunity at a given point (x, y) is calculated as:

$$missed\ opportunity\ =\ possible\ score\ -\ actual\ score$$

It represents the amount of coursework that the student has missed so far. The relative achievement on the other hand is calculated as:

$$relative\ achievement\ = \frac{actual\ score}{possible\ score} \times 100$$

It represents how much of what is possible for the student to achieve is actually achieved.

Figure 2 shows how these three features behave over time for two passing and struggling students.

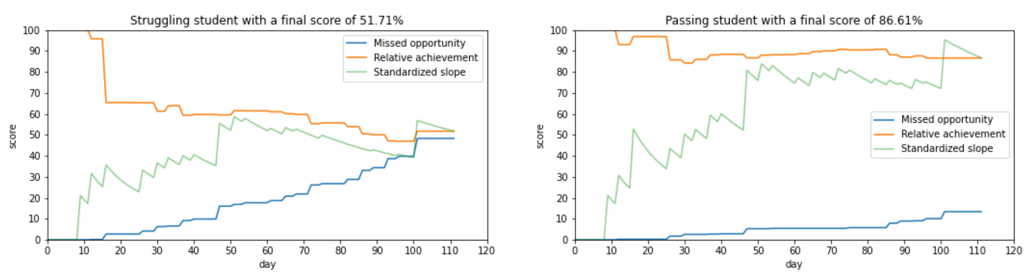

Figure 2: Missed opportunity, relative achievement, and standardized slope for two students

Do passing or failing students have similar curves? We can use the aforementioned features to visualize the differences between these two groups of students. Figure 3 shows what these features look like for three passing students from an Object-Oriented Programming course with different final passing grades.

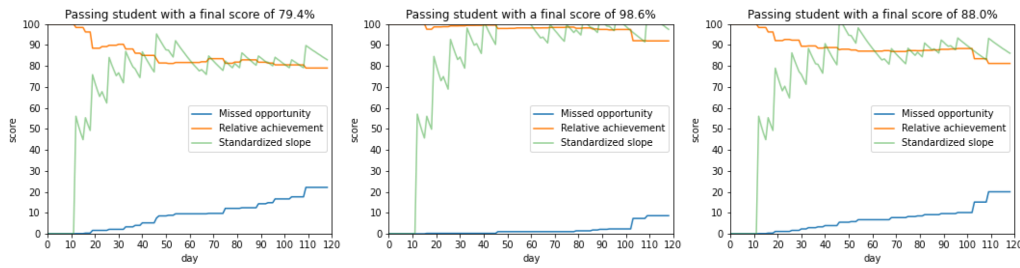

Figure 3: Example passing students

Similarly, Figure 4 shows the same features for three struggling students from the same course with different final failing grades.

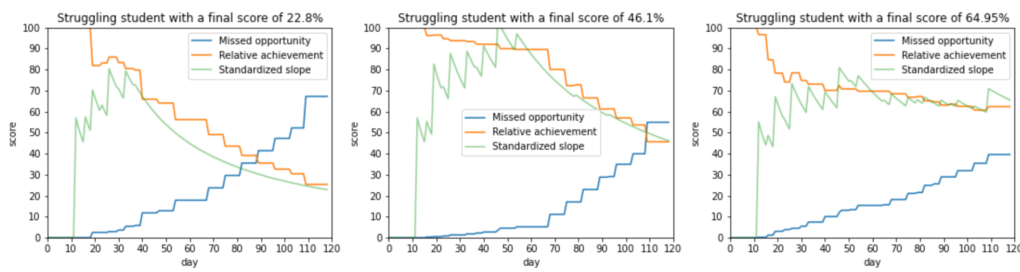

Figure 4: Example struggling students

Comparing these two figures, one can see that:

- The relative achievement curve is flatter and closer to a line in the case of passing students than it is for struggling students.
- The missed opportunity curve grows much faster and higher for struggling students than it does for passing students.
- While the standardized slope curve looks similar for both groups of students at the beginning of the course. It stabilizes for passing students after the first third of the course. For struggling students, however, it declines and continues to do so till the end of the course, indicating that the student either has given up or is still making attempts that are not enough or too late to change the outcome of the course.
- Overall the plots for struggling students are more diversified than those for passing students. This suggests that students struggle through courses in different ways and those differences show in these plots.

Furthermore, we can roughly summarize the ups and downs of the standardized slope curve from the beginning of a course to a given time point using a regression line. Figure 5 depicts two regression lines summarizing the standardized slopes of two students. To make this line fit the curve better, the few zero slope values at the beginning of the course (when nothing has been submitted for grading) are treated as outliers and ignored. These regression lines represent the trends of students' progress through courses.

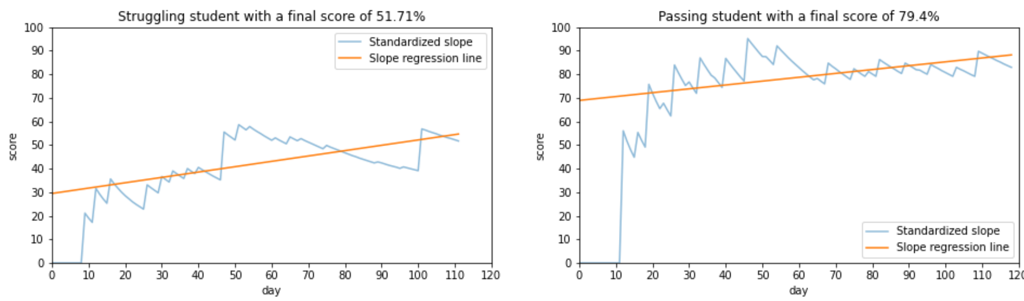

Figure 5: Standardized slope curves and corresponding regression lines

In addition, a regression line at a time point would summarize the standardized slope curve from the beginning of the course up to that point. Doing this allows us to visualize the variability in student progress that exists between two time points. For instance, the shaded areas of Figure 6 represent this variability between three different time points (10%, 33%, and 67% of the course) and the end of the course.

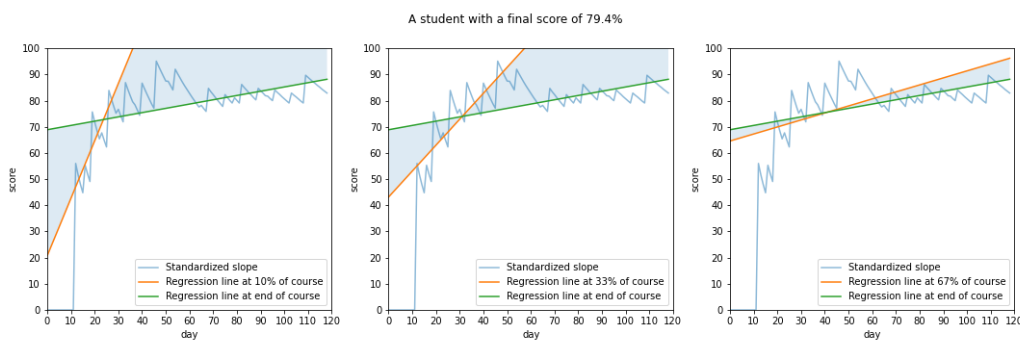

Figure 6: Student performance variability at three different time points

As Figure 6 shows, there is more variability between the 10% point of the course and the end of the course than there is between the 33% point and the end of the course. These variabilities can be thought of as indicators of the student's ability to change the outcome of the course during the time between that point and the end of the course. As can also be seen from Figure 6, at the 67% point of the course little variability remains, which indicates little and/or too late ability for a struggling student, for example, to change the outcome of the course.

The missed opportunity, relative achievement, and standardized slope features can be calculated at every single time point for every student. The result would be a large dataset that can be used to build and evaluate supervised machine learning models that accurately predict student performance at the end of a course from any given time point during that course. Each student contributes to this dataset with as many data examples as the number of time points (days) within the span of the course they are taking.

Results

This section describes how the predictor features of the dataset of the last section are used to build and evaluate supervised machine learning models that accurately predict student outcomes at the end of the course. As mentioned before, this is a binary classification problem with two classes: failed (final score < 74%) and passed (otherwise).

The dataset developed in the previous section is a collection of time series each corresponding to a student in a course. While there are standard methods for time series forecasting (both univariate and multivariate)[9], these time series do not lend themselves well to these forecasting methods. First, a single time series corresponds to a single student within a single course and does not have enough history to warrant the use of these methods. Secondly, this study looks at this data at the course level, and that makes it hard to define a meaningful seasonality for these time series. Because of these reasons, the remainder of this section focuses on applying supervised machine learning models to this dataset.

It is imperative to remember here that the goal is to be able to perform student outcome predictions at any given time point during a course. To start, we passed the dataset of the previous section along with a time point to a function that looks at the individual time series and returns the values corresponding to that time point. In other words, the returned dataset of this function is a typical machine learning dataset where every student is represented by a single data example and the values of that example depend on the given time point.

Next, we run a few data experiments to determine which of the three aforementioned predictor features result in better predictions. These data experiments show that the relative achievement and missed opportunity are individually better predictors than standardized slope. The combination of these three features, however, gives the best results. As a result of that, all of these three predictor features will be used in the remainder of this section.

We then train and evaluate various supervised machine learning models at different time points during the three courses (each with multiple sections) involved in this study. We used the Scikit Learn[8] Python library for that purpose. While we evaluated multiple models; for brevity, this section reports the outcome of only one classifier (the Gaussian Naive Bayes) trained and tested separately on three datasets (one for each course). This is also because the output of the other evaluated models are similar. Figure 7 shows the results of the Gaussian Naive Bayes classifiers when trained and tested on these datasets.

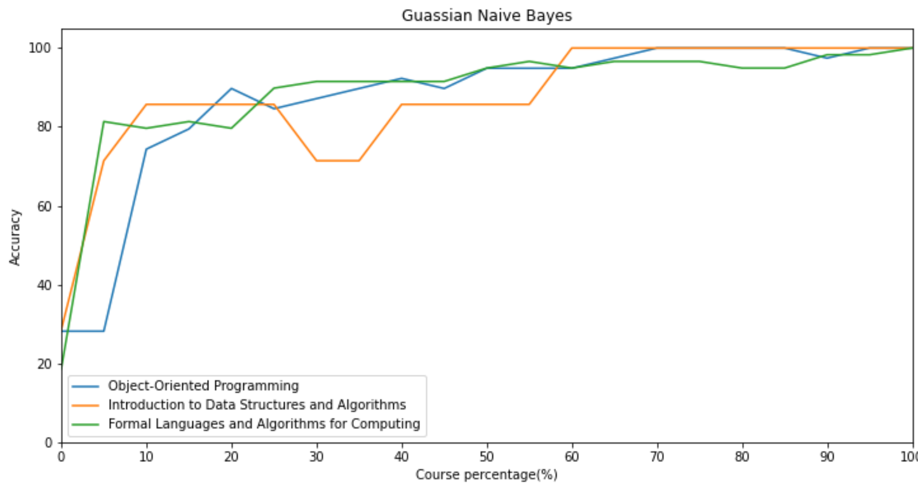

Figure 7: Performance of Gaussian Naive Bayes on datasets from three courses

The x-axis represents the course time points as percentages. The y-axis represents the accuracy of predicting whether students will pass or fail at the end of the course. The results of Figure 7 are obtained by a 75-25% training-test data split. The model is trained on the training set and evaluated on the unseen test set. No cross-validation is used here.

The datasets used in the results of Figure 7 are unbalanced. The number of students who passed in these datasets are much more than those who failed. Having an unbalanced dataset affects the usefulness of the prediction accuracy as a measure of model performance. To fix this problem, we experimented with the following dataset balancing strategies:

- Making the number of passed data examples the same as the number of failed data examples by synthesizing new data examples with new ids based on existing failed examples. Here we randomly remove submissions from copied failed examples and recalculate the score totals.
- Making the number of passed data examples the same as the number of failed data examples by synthesizing new data examples with new ids based on existing passed examples. The idea here is to randomly remove enough submissions from copied passed examples until they fail and recalculate the score totals.
- Balancing the data by randomly removing extra passed data examples. This results in a reduced-sized dataset, which requires the use of cross-validation.

The first two strategies did not work well with any model. Part of the reason is that the synthesized examples are not realistic and, as a result, make it difficult for the evaluated models to learn. They also lead to unstable models that work well at some time points and badly at others.

Despite being seemingly wasteful, the last balancing strategy, coupled with LOOCV (Leave-One-Out-Cross-Validation) yields the best results. Figure 8 shows the performance of Gaussian Naive Bayes on balanced datasets using this strategy.

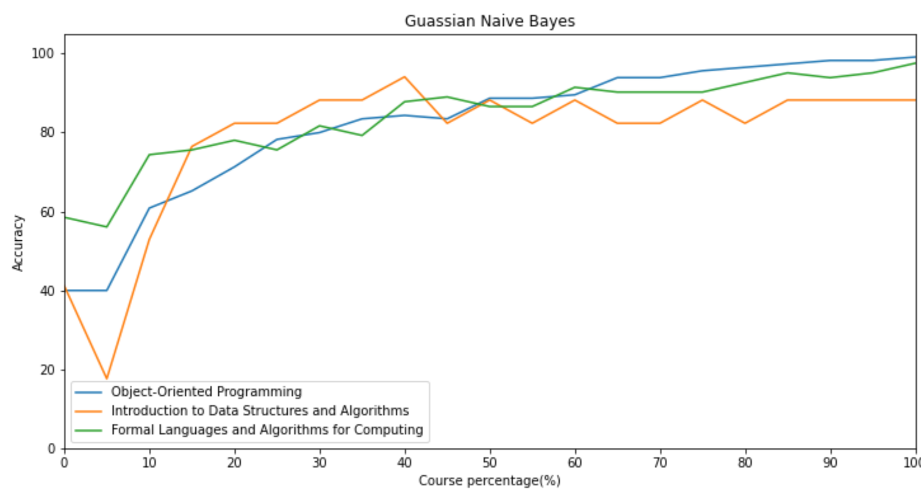

Figure 8: Performance of Gaussian Naive Bayes on balanced datasets using LOOCV

Figures 7 and 8 show that the relative achievement, missed opportunity, and standardized slope features are useful predictors of student performance outcome. Using a model such as the one depicted in these figures, at-risk students (those predicted to fail at the end of the course) can be identified early enough to intervene and provide help. For example, Figure 8 shows that at the 30% point of any of these three courses, one can predict with about 80% accuracy who will fail and who will pass these courses. As more time passes, the accuracy of the model increases. Furthermore, the accuracies reported in these figures are similar to, if not better than, what has been reported in similar studies[10]. In addition, the fact that these are three different courses taught by three different instructors, indicates that these predictor features are robust enough to produce similar results across multiple courses and instructors.

Future work and conclusion

In addition to what has been presented in this paper, the other overarching questions of this study are:

- Is course modality a significant factor? How do models trained on a course section of a specific modality predict the performance of students of another course section by the same instructor using a different modality?
- Is the instructor a significant factor? How do models trained on a section by one instructor predict the performance of students of another section of the same course taught by a different instructor?
- Are there global patterns that repeat across multiple courses, sections, instructors, and modalities?

The last question is particularly interesting because it allows us to investigate whether students' learning experiences within a course are unique to students, similar within groups/categories of students, or if they are patterns that transcend students and courses. All of these questions remain of interest to this work-in-progress study.

More work is also needed to augment the current supervised models using other LMS data. Research has shown, for instance, that activity counts, by themselves, are not good predictors of student performance[2]. Finding whether incorporating these counts into the models described in this paper improves their performance is yet to be investigated.

In addition, the portability of model predictions across courses and instructors is an issue that has been pointed out in research[4]. While this study has shown that results can be similar across multiple courses taught by different instructors, more investigation into the same courses taught by different instructors is still needed. Further investigation is also needed into whether it is useful to have a dataset consisting of multiple courses in different modalities taught by different instructors. Doing so will show whether certain patterns of students' progress through courses can transcend courses, modalities, and instructors.

Finally, the scope and focus of this paper prevents, for reasons listed under the approach section, from applying standard time-series forecasting methods to course-level student time series. Compare that to a study tracking the progress of students across multiple consecutive courses. In this case, one can think of the course as the seasonality of these time series, which will also have enough history to apply these forecasting methods.

In summary, this paper explores the use of LMS data related to assignments and other graded activities in gaining insights into how students advance through courses. It describes a time-based approach to using this data to predict student performance outcomes at the end of the course from any given time point during the course. The ultimate goal is to be able to use such predictions to implement early intervention measures and improve 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] 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.

[9] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles and Practice, Otexts, 2021.

[10] E. Alyahyan and D. Düştegör, "Predicting academic success in higher education: literature review and best practices," *International Journal of Educational Technology in Higher Education,* vol. 17, no. 1, p. 3, 2020.