Check for updates

Ultra-Lightweight Early Prediction of At-Risk Students in CS1

Chelsea Gordon
zyBooks
Campbell, California
chelsea.gordon@zybooks.com

Stanley Zhao
Computer Science & Engineering
University of California, Riverside
Riverside, California
szhao050@ucr.edu

Frank Vahid
Computer Science & Engineering
University of California, Riverside
vahid@cs.ucr.edu
Also with zyBooks

ABSTRACT

Early prediction of students at risk of doing poorly in CS1 can enable early interventions or class adjustments. Preferably, prediction methods would be lightweight, not requiring much extra activity or data-collection work from instructors beyond what they already do. Previous methods included giving surveys, collecting (potentially sensitive) demographic data, introducing clicker questions into lectures, or using locally-developed systems that analyze programming behavior, each requiring some effort by instructors. Today, a widely used textbook / learning system in CS1 classes is zyBooks, used by several hundred thousand students annually. The system automatically collects data related to reading, homework, and programming assignments. For a 300+ student CS1 class, we found that three data metrics, auto-collected by that system in early weeks (1-4), were good at predicting performance on the week-6 midterm exam: non-earnest completion of the assigned readings, struggle on the coding homework, and low scores on the programming assignments, with correlation magnitudes of 0.44, 0.58, and 0.72, respectively. We combined those metrics in a decision tree model to predict students at-risk of failing the midterm exam (<70%, meaning D or F), and achieved 85% prediction accuracy with 82% sensitivity and 89% specificity, which is higher than previously published early-prediction approaches. The approach may mean that thousands of instructors already using zyBooks or a similar system can get a more accurate early prediction of at-risk students, without requiring extra effort or activities, and avoiding collection of sensitive demographic data.

CCS CONCEPTS

• Social and professional topics - Professional topics - Computing education - Computing education programs - Computer science education - CS1.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada.
© 2023 Association of Computing Machinery.
ACM ISBN 978-1-4503-9431-4/23/03...$15.00.
https://doi.org/10.1145/3545945.3569764.

KEYWORDS

CS1, early predictors, computer science education, grade predictors, earnestness, struggle, early intervention

ACM Reference Format:

Chelsea Gordon, Stanley Zhao, and Frank Vahid. 2023. Ultra-Lightweight Early Prediction of At-Risk Students in CS1. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023), March 15-18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3545945.3569764.

1 INTRODUCTION

Fail rates in college-level introductory programming courses (aka CS1) are known to be 25% to 33% [17], with rates being even higher at some institutions. CS instructors have long sought early-term predictors of students at-risk of doing poorly, to enable intervention [15, 31] or to make class adjustments.

One such method of identifying at-risk students is to build a predictive model for student success. Several research studies have attempted such an approach to predict performance on standardized tests [2], final course grades [29, 8, 7, 12, 26, 21, 31], final exams [15], online course performance [4], and completion of MOOC courses [13, 23, 25, 31]. These studies use data based on behavioral [29, 15, 12, 13, 23, 25], previous academic performance [2], demographic [26, 20] and/or self-reported [21, 16, 2, 31] features as their predictive measures.

One study presented a model that achieved 64% accuracy on whether students would fail, pass, or excel, using time spent on and between assignments, and time before the due date assignments were submitted [28]. Another study achieved 71-80% accuracy using a Decision Tree model using performance data from the ongoing course [1], but used sensitive student data and student surveys in addition to behavioral measures. Another used principal components analysis (PCA) with clicker data to build a model of nine latent factors that successfully predicted pass or fail on the final exam for 70% of students [15]. While a promising result, PCA may make it harder to gain actionable insights to encourage interventions.

Many approaches have required instructors to collect extra data from students, such as via in-lecture clickers (which students must purchase), additional in-class activities, and/or surveys (and, sometimes such survey data is sensitive, like demographic data including a person's sex or gender, race, financial status, etc). Some approaches have used programming systems to collect programming behavior data, but with those systems being local to an institution and not widely available to other instructors.

In the past few years, several cloud-based learning systems have evolved for use in CS1 courses, such as Runestone Academy [24], CodeHS [3], Codio [5], PrairieLearn [30], and more. Another is zyBooks [33], which one report indicates is the most widely used [18], used in over 2,000 courses annually [10]. Their system includes an online textbook, homework system, and program development/auto-grading environment. We use the zyBooks system in our CS1 that serves about 1,500 students per year. As such, we sought to determine if the data automatically available in their system, requiring no extra instructor effort, could predict student performance. If so, thousands of courses would have an option to detect at-risk students, and potentially intervene if desired. While some previous works refer to "lightweight" techniques due to using items like surveys or clicker data, we refer to our approach as "ultra-lightweight" since it requires *no* effort to collect any data beyond what is already automatically available from zyBooks.

We were most interested in *early* detection, to enable early intervention. We thus focused on data from the first few weeks of the term. Furthermore, knowing that doing poorly on the midterm leads to withdrawals or to low course grades, we mostly focused on predicting midterm scores, whereas most previous work focused on predicting course grade. A previous study [31] focused on midterm scores as well, developing a model to predict midterm scores using student survey responses related to comfort level, math skills, and attribution to luck for success. The model achieved an R^2 value of 0.4443, which is a moderate effect size.

We examined numerous data metrics and eventually narrowed in on three metrics that had strong predictive ability: earnestness, lab completion, and challenge activity struggle. This paper describes our course, the various metrics and their midterm exam predictive capability, and a decision tree model combining the metrics to predict students at-risk of failing the midterm exam.

2 OUR CS1 COURSE

Our CS1 at UCR serves ~1500 students per year, about half computing majors, and the other half science and engineering majors. The class is offered every quarter for 10 weeks via 3-5 ~100 student sections, plus summer sections. The class is taught by experienced instructors, has strong course evaluations, good grades, and yields solid student performance in CS2 and CS3. The class uses pedagogical approaches known to aid student success: flipped lectures, active learning, scaffolding, many-small-programs, auto-grading, peer instruction, collaboration, growth mindset, help normalization and resources (learning assistants, supplemental instructors, office hours, real-time discussion forum).

Each week, students in the zyBook read and answer ~100 learning questions (Participation Activities or PAs), complete ~20 homework code reading and writing problems (Challenge Activities or CAs), and code 5-10 programming assignments (Lab Activities or LAs), designed with the expectation of roughly 7-9 hours per week for students with no prior experience. Students are required to do all programming in the zyBook (no external tools allowed), to reduce cheating, reward effort, and enable analysis. We configure our book so each chapter is one week. All zyBook activities are auto-graded, with immediate feedback, partial credit, and resubmissions. Instructors can download reports of activity completion, and of time divided into PA, CA, and LA time. That time represents most student time spent outside scheduled lecture time. According to zyBooks, most courses award students points for completing PAs, CAs, and LAs [6].

Our CS1 course grade consists of ~10% PAs, ~10-15% CAs, ~15-20% LAs, ~5-10% class participation, and ~50-60% in-person proctored exams. The high exam weight enables gentler policies on other items, like allowing collaboration (within bounds). The exams are half multiple choice, and half code writing graded by TAs on Gradescope using a detailed instructor-provided rubric, requiring dozens of hours, overseen by instructors for quality.

3 METRICS

3.1 Week 1-4 earnestness

zyBooks PAs include watching animations and answering interactive learning questions. Most of the learning questions are multiple choice, but some require typing a short answer in a text box. Short answer questions include a clickable "Show answer" button, which reveals the correct answer. Students may then type (or copy-paste) that answer into the answer text box to receive full credit for the question.

Most students earnestly complete the PAs, while a fraction rush through. Short answer questions are a strong indicator of earnestness. We define an "earnestness" metric as the proportion of short answer questions that students earnestly attempted to complete. An earnest attempt is when a student attempts to provide a response to the question, whether incorrect or correct, *prior* to clicking "show answer". A blank answer does not count as an earnest attempt. In our class, most students complete most short answer questions earnestly (as will be seen in Figure 1), but some repeatedly use the "Show answer" button without really trying.

For this metric and the metric discussed further below, we analyze up to Week 4, because we want *early* prediction that could potentially enable intervention before our midterm given at the start of Week 6. We thus compute a Weeks 1-4 earnestness metric score as the proportion of earnest attempts to total attempts in weeks 1-4, multiplied by the overall proportion of short answer questions answered (to normalize for completion). We later include some analysis for Weeks 1-3, and Weeks 1-2.

This metric may measure how disciplined the students are in trying to learn, versus rushing through to get points. (In some cases though, low earnestness is due to students not needing to do the readings due to having prior programming experience).

3.3 Weeks 1-4 challenge activity (CA) struggle

zyBooks include Challenge Activities (CAs) at the end of most sections. A CA represents a small homework problem, each requiring perhaps 1-4 minutes, involving typing the output of a small program, or typing code to complete a small program to achieve a task, such as "Complete the program to output the maximum of the three input integers". We define CA struggle using two parameters: number of attempts (number of submissions), and time spent trying to solve the CA. Students may submit multiple submissions before achieving the correct solution to a CA. We define a struggling student for a particular CA as a student who has spent more than twice the average time and attempted more than double the average attempts.

We define a Weeks 1-4 CA struggle metric score as the proportion of total CAs that a student had struggled on in chapters 1-4.

This metric may measure whether students are understanding the concepts and developing needed programming skills. Struggle may come from various factors, such as not reading the preceding sections carefully / earnestly, trying to rush through the CAs, etc.

3.2 Weeks 1-4 lab completion

zyBooks includes a program auto-grader known as zyLabs for their Lab Activities (LAs). The zyLab tool includes a simple integrated development environment (IDE) in which students can write and run programs. Students can provide input and observe the program's output. The lab tool's auto-grader automatically tests a submitted program for input/output correctness against various test cases, or can run unit tests (calling a student's function/method directly and checking the returned results). Students are immediately shown their current score, which may include partial credit if passing some test cases but failing others. Students can resubmit multiple times without penalty.

We defined a weeks 1-4 lab completion metric score as the completion score achieved on LAs in weeks 1-4.

This metric measures how well students are doing on the main programming assignments in the course, which is clearly an important aspect of the course.

4 CORRELATIONS

We evaluated data for our CS1 offering from Spring quarter 2022, involving 265 students. We considered "early" data, namely weeks 1-4. (Later, we provide info for Weeks 1-3 and 1-2).

A linear regression model was fitted to the data. The dependent variable was midterm grade, and the independent variables were earnestness, challenge activity struggle, and lab completion. The model reached significance overall ($F(3,265) = 187.9$, $p < .0001$, R-squared = .68). T-tests using Satterthwaite's method revealed a significant effect of earnestness ($t = 3.575$, $p < .0001$, $\eta p2 = .05$), struggle ($t = -9.695$ $p < .0001$, $\eta p2 = .26$), and lab completion ($t = 14.414$, $p < .0001$, $\eta p2 = .44$).

Figure 1 shows midterm score versus week 1-4 earnestness. Each dot is a student. A trendline is shown for convenience. The students along the bottom did not take the midterm. Earnestness has a moderate effect size (partial eta-squared ($\eta p2$) = .05). The more earnestly students complete the activities, the better they perform on the midterm exam. Figure 2 shows midterm score versus week 1-4 CA struggle. CA struggle has a large effect size (partial eta-squared ($\eta p2$) = .26). The more CAs that students struggle on, the worse they perform in the midterm. Figure 3 shows midterm score versus week 1-4 lab completion. Lab completion has a very large effect size (partial eta-squared ($\eta p2$) = .44). The more labs students complete, the better they perform on the midterm exam.

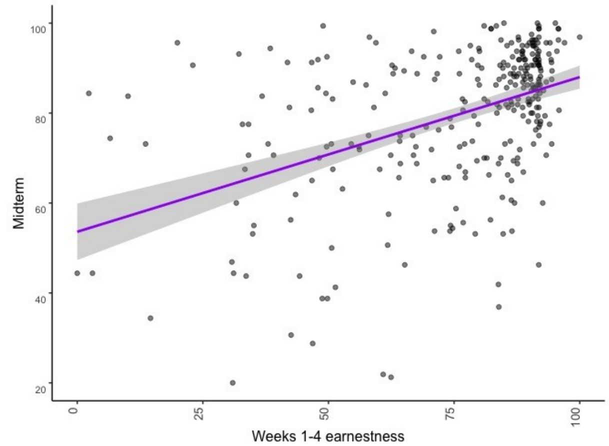

Figure 1: Midterm scores vs. earnestness. The raw Pearson correlation is 0.44.

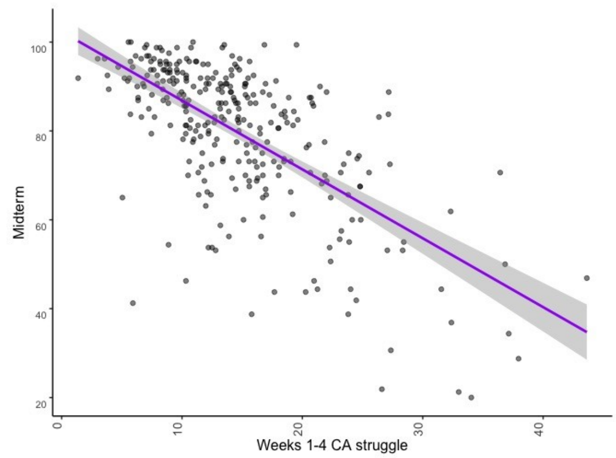

Figure 2: MT scores vs. CA struggle. Correlation: -0.58.

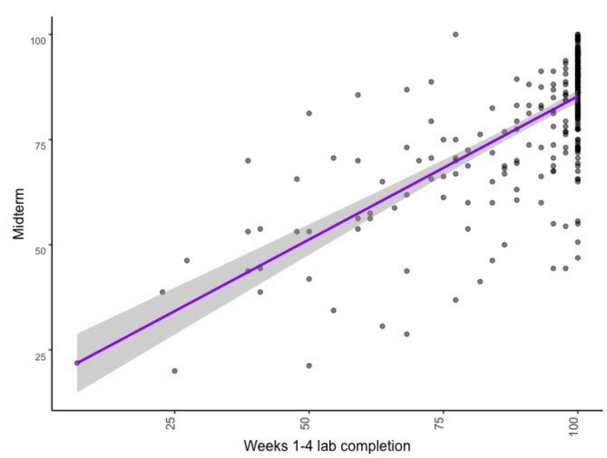

Figure 3: MT scores vs. lab completion. Correlation: 0.72.

As a general reminder, this data only shows correlations; the metrics might be direct causal factors, or other factors may explain the correlations.

5 MIDTERM EXAM DISTRIBUTIONS

We were interested to see grade distributions for various levels of each metric. Figure 4 shows distribution of midterm exam grades for different bins of earnestness levels. The midterm score cutoffs for A/B/C/D were 90/80/70/60. As can be seen, students in the 26-50% earnestness bin were more likely to receive a D or F, and students in the <25% bin mostly received Ds and Fs. In contrast, students in the >75% bin mostly received As and Bs, and rarely received a D or F.

Figure 5 shows midterm grade distributions for levels of early struggle. Students with high struggle (in the left bins) mostly received Cs, Ds, and Fs. Students with little struggle mostly received As and Bs.

Figure 6 shows midterm grade distributions for levels of lab completion. Nearly all students in the lower two bins received Fs or Ds. Interestingly, the only bin with As on the midterm were students with at least 75% lab completion.

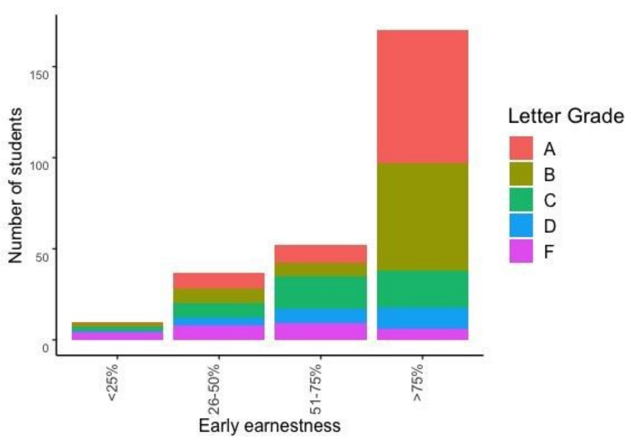

Figure 4: MT grade for earnestness bins.

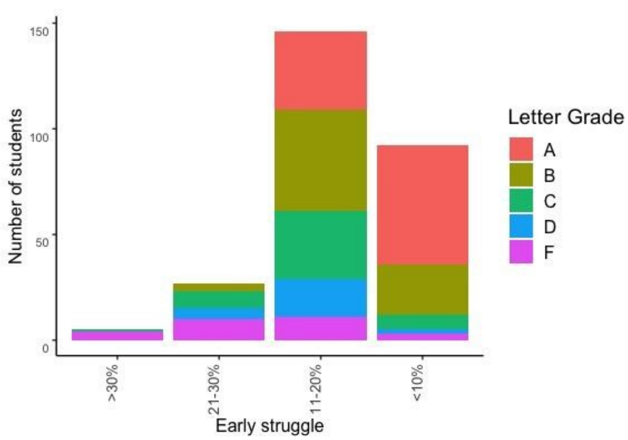

Figure 5: MT grade for CA struggle bins.

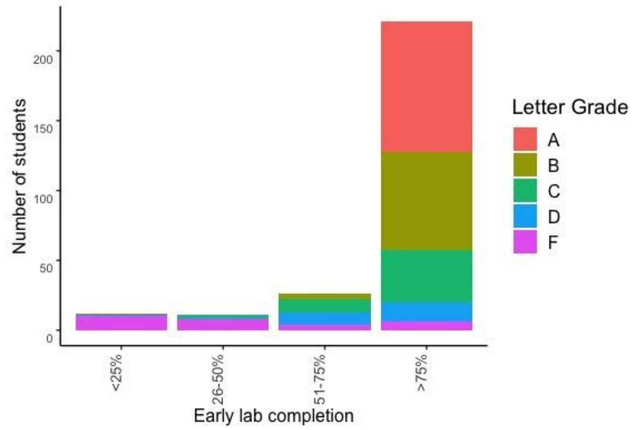

Figure 6: MT grade for lab completion bins.

6 ANALYSIS

We sought to combine these correlating metrics in a single model that could predict students at risk of "failing" our midterm exam. Our first model was a multiple regression model using all three variables to predict midterm score. The model reached significance ($F(3, 292) = 205.8$, $p < .0001$) and had an adjusted R^2 of .675. We then split our data into 70% training data and 30% test data, and trained the model to predict midterm score. The Root Mean Squared Error (RMSE) for the prediction versus actual score was 8.34. We then sampled randomly from the same distribution 1,000 times and observed an average RMSE of 20.78, confirming that our prediction model does substantially better than chance. Figure 7 shows the estimated versus actual scores using this model.

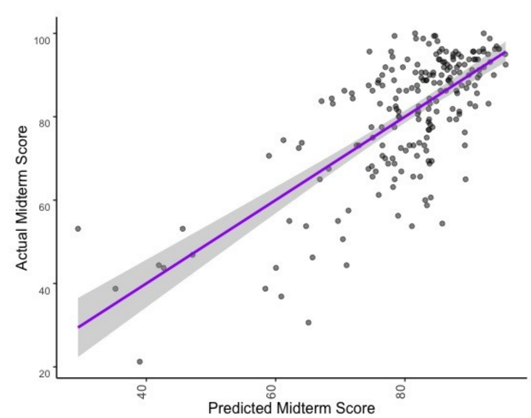

Figure 7: Actual midterm score regressed onto the predicted score generated by our linear model.

Because we were interested in predicting whether students were at-risk or not, we then applied a Decision Tree model for classifying students as either at-risk (<70% on the midterm exam, meaning a D or F) or not at-risk using the three above specified predictors. The data were split into 70% training data, and 30% test data. We ran the model 1,000 times with different randomizations of the training/test data selected. The average accuracy from these 1,000 runs is the accuracy metric we report. The decision tree model performed at 89% accuracy on the training data, and 85% accuracy on the test data. The sensitivity of the model is 82% and the specificity is 89%. In other words, this model accurately identified 82% of at-risk students, and accurately identified 89% of students who were not at risk.

For interest, we show an example generated decision tree in Figure 8. If we follow the nodes, the decision tree rules are as follows:

1. If lab completion < 65%, student is at-risk; else:
2. If lab completion >= 96%, student is not at-risk; else:
3. If struggle < 13%, student is not at risk; else:
4. If earnestness >= 68%, student is not at risk; else: student is at risk

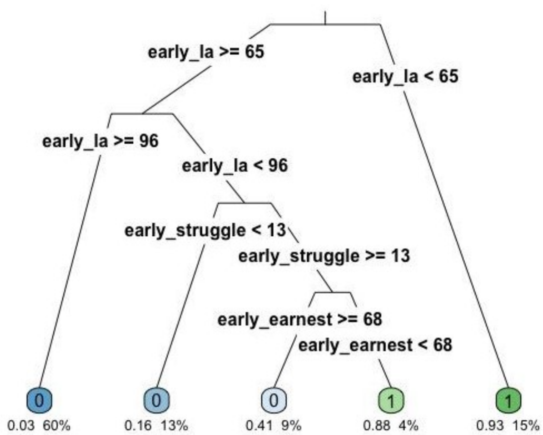

Figure 8 - The decision tree model for classifying students as at-risk (1) or not at-risk (0).

We were also interested in determining how accurate our predictions would be when using the data available only in earlier weeks. We found that a weeks 1-2 model achieved 78% accuracy, weeks 1-3 achieved 80% accuracy, vs weeks 1-4 achieving 85% accuracy. Interestingly, weeks 1-5 accuracy did not increase, instead coming in at 84%. In short, instructors can begin getting reasonably accurate early prediction of at-risk students with even just two weeks of data (and thus begin interventions), but waiting another week or two improves the accuracy.

We explain here why we focus on predicting the midterm exam score rather than course grade (which we also analyzed). The midterm is closed-book and carefully proctored, and thus is a strong measure of student learning absent help from classmates, tutors, or online resources. We have observed that students who do poorly on the midterm often do poorly in the course overall, even dropping the course after a bad midterm performance. Figure 9 shows the total course score versus the midterm exam score, where the total course score had 110 possible points with score cutoffs for A/B/C/D being 90/80/70/60. The red box shows that most students who scored <70% on the midterm (x axis) ended with <70% in the course. (For readers wondering about the students on the y axis, those are students who did not take the midterm. Most dropped the course before the midterm, but a few students missed the midterm (e.g., due to illness) but still completed the course. The red points in this plot illustrate students who were predicted at-risk in our Decision Tree Model.

We note that students who perform poorly in the midterm are not always the same students who are at-risk in the course overall. While there is a strong correlation, some students might perform well in the midterm, but not in the overall course, in Figure 9's orange box. These include students who struggle with material later in the course. We explored how our Decision Tree model would perform when trained and tested on students who were at-risk (<70%) in the course overall. The model still reached 84% accuracy for predicting course grade, but might reach even higher if we take into account performance after week four. Future work aims to examine how we can improve the prediction for overall course grade using additional behavioral data past week four.

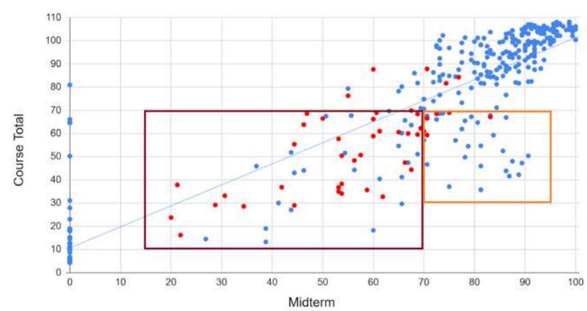

Figure 9 - Why we focus on midterm score: Midterm score correlates strongly with course grade (R-squared = 0.781). Students getting a D or F on the midterm (<70%) almost all get a D or F in the course, as shown via the red box. Some students, however, score >= 70% on the midterm but still do poorly in the course (orange box) -- an area for future investigation. The red points illustrate students who were predicted as at-risk.

7 THREATS TO VALIDITY / DISCUSSION

The data shows correlation, not causation. One might imagine earnestness could have a causal relationship with midterm score; non-earnest students might be skimming the material and thus not learning. However, the relationship could be due to a common third factor, such as a student who is not really trying to do well in the course. Likewise for CA struggle and lab completion.

We only fit the model to data from sections in one particular CS1 course. Because our model was built using only 265 students at one university, it may not generalize to other courses. Rather, the model should be fitted with the actual data for other courses using this tool. Future studies might determine whether a general model subsuming multiple courses is useful.

This research may not generalize across disciplines or across course levels. The model used in this context performed very well when predicting CS1 midterm scores and at-risk students. It is quite possible, however, that the model would perform worse in a more advanced CS course, or in a different discipline.

Past programming experience was surveyed and considered for our model, but was not found to have a significant impact on midterm exam performance, contrary to what some other studies have shown [27, 22]. One reason might be that this course used zyBooks, which were specifically designed to promote active learning with scaffolding and extensive feedback, thus helping those with less prior experience to do well in the course.

As such, the lack of prior programming significance in our model might be less likely to replicate in courses using other courseware.

9 INTERVENTIONS

A key principle of this and related early-prediction research is that detecting at-risk students early enables interventions that can improve outcomes. In previous work, interventions start at one-fourth to one-third of the way into the class and may continue to the end [19]. Our model can generate accurate predictions of a week 6 midterm score from the first few weeks of our 10-week course.

Past work on interventions in introductory level college classes have found that a "soft nudge" approach, which includes offering course-specific tutoring and better time management practices, resulted in a 6.5 to 7.5 point increase for students just below the intervention threshold compared to students just above the threshold [9]. Contacted at-risk students who chose to opt in for additional advising saw a higher pass rate (57.7%) when compared to peers who did not opt in (33.3%) [32]. We note that such previous research is subject to self-selection bias, where contacted students choose to participate in the interventions.

Previous studies for interventions in CS1 courses in particular have used techniques such as metacognitive interventions and mindset interventions [14], specialized seminars as an alternative to lectures [11], or scheduled one-on-one meetings. These studies have reported low to mild improvement. An interesting note regarding interventions is to take care to avoid negative consequences, such as demotivating at-risk students ("the professor thinks I'm going to fail, maybe I should just quit" or "hmm, I got contacted but my friends didn't, maybe I'm not cut out for this").

Our hope is that with accurate early predictors of at-risk students as targeted in this work, available on a much larger scale than previously possible (due to being immediately available in a widely-used existing CS1 learning system), more intervention approaches can be experimented with, in a more focused way, to enable successful early interventions in CS1 courses leading to improved outcomes.

10 CONCLUSIONS

This work proposes an ultra-lightweight method of predicting midterm exam performance (and course performance as well) early in a term to enable early detection of at-risk students and thus potential intervention. Examining weeks 1-4, we found lab completion to have the strongest correlation (0.72), followed by homework struggle (-0.58), and earnest textbook reading (0.44). A decision tree model using those metrics could predict students at-risk of getting a D or F on the midterm with 85% accuracy (and a D or F in the course with 84% accuracy). Not only is such accuracy higher than in previous research, but it is attained with no effort by instructors already using zyBooks and does not require collecting personal data or taking extra time in lecture to collect data. As such, this early data may prove useful to large numbers of instructors who might apply early-proven intervention techniques or try new intervention techniques, as we hope to do in our future CS1 offerings.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 2111323.

REFERENCES

[1] Ahadi, A., Lister, R., Haapala, H., & Vihavainen, A. (2015, August). Exploring machine learning methods to automatically identify students in need of assistance. In Proceedings of the eleventh annual international conference on international computing education research (pp. 121-130).

[2] Alspaugh, C. A. (1972). Identification of some components of computer programming aptitude. Journal for Research in Mathematics Education, 89-98.

[3] CodeHS, https://codehs.com/, accessed 2022.

[4] Campbell, J., Horton, D., & Craig, M. (2016, July). Factors for success in online CS1. In Proceedings of the 2016 acm conference on innovation and technology in computer science education (pp. 320-325).

[5] Codio, https://www.codio.com/, accessed 2022.

[6] Edgcomb, A. and Vahid, F., 2015, October. How many points should be awarded for interactive textbook reading assignments?. In 2015 IEEE Frontiers in Education Conference (FIE) (pp. 1-4). IEEE.

[7] Fire, M., Katz, G., Elovici, Y., Shapira, B., & Rokach, L. (2012). Predicting student exam"s scores by analyzing social network data. In Active Media Technology (pp. 584-595). Springer Berlin Heidelberg. [7] Fowler, G. C., & Glorfeld, L. W. (1981). Predicting aptitude in introductory

[8] Fowler, G. C., & Glorfeld, L. W. (1981). Predicting aptitude in introductory computing: A classification model. AEDS Journal, 14(2), 96-109.

[9] Gordanier, J., Hauk, W., & Sankaran, C. (2019). Early intervention in college classes and improved student outcomes. Economics of Education Review, 72, 23-29.

[10] Gordon, C., Lysecky, R. and Vahid, F., 2021. The rise of the zyLab program auto-grader in introductory CS courses. zyBooks.com, White Paper.

[11] Harrington, B. (2017, May). Get On Track, Stay On Track: Combining Early Intervention and Study Skills in a CS1 Seminar. In Proceedings of the 22nd Western Canadian Conference on Computing Education (pp. 1-1).

[12] Jiang, S., Warschauer, M., Williams, A. E., O"Dowd, D., & Schenke, K. (2014). Predicting MOOC Performance with Week 1 Behavior. In Proceedings of the 7th International Conference on Educational Data Mining.

[13] Kloft, M., Stiehler, F., Zheng, Z., & Pinkwart, N. (2014). Predicting MOOC Dropout over Weeks Using Machine Learning Methods. EMNLP 2014, 60

[14] Krause-Levy, S., Porter, L., Simon, B., & Alvarado, C. (2020, February). Investigating the impact of employing multiple interventions in a cs1

course. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (pp. 1082-1088).

[15] Liao, S. N., Zingaro, D., Laurenzano, M. A., Griswold, W. G., & Porter, L. (2016, August). Lightweight, early identification of at-risk CS1 students. In Proceedings of the 2016 acm conference on international computing education research (pp. 123-131).

[16] Liao, S. N., Valstar, S., Thai, K., Alvarado, C., Zingaro, D., Griswold, W. G., & Porter, L. (2019, July). Behaviors of higher and lower performing students in CS1. In Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education (pp. 196-202).

[17] Luxton-Reilly, A., Ajanovski, V. V., Fouh, E., Gonsalvez, C., Leinonen, J., Parkinson, J., Poole, M., and Thota, N. (2019). Pass rates in introductory programming and in other stem disciplines. In Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education (pp. 53-71).

[18] Navstem CS1 report, https://navstem.com/, 2021.

[19] Quille, K., & Bergin, S. (2019). CS1: how will they do? How can we help? A decade of research and practice. Computer Science Education, 29(2-3), 254-282.

[20] Quille, K., & Bergin, S. (2018, July). Programming: predicting student success early in CS1. a re-validation and replication study. In Proceedings of the 23rd annual ACM conference on innovation and technology in computer science education (pp. 15-20).

[21] Rountree, N., Rountree, J., & Robins, A. (2002). Predictors of success and failure in a CS1 course. ACM SIGCSE Bulletin, 34(4), 121-124.

[22] Rountree, N., Rountree, J., Robins, A., & Hannah, R. (2004). Interacting factors that predict success and failure in a CS1 course. ACM SIGCSE Bulletin, 36(4), 101-104.

[23] Rosé, C. P., & Siemens, G. (2014). Shared task on prediction of dropout over time in massively open online courses. EMNLP 2014, 39.

[24] Runestone Academy, https://runestone.academy, accessed 2022.

[25] Sharkey, M., & Sanders, R. (2014). A Process for Predicting MOOC Attrition.EMNLP 2014, 50.

[26] Ventura Jr, P. R. (2005). Identifying predictors of success for an objects-first CS1.

[27] Ventura, P., & Ramamurthy, B. (2004). Wanted: CS1 students. No experience required. ACM SIGCSE Bulletin, 36(1), 240-244.

[28] A. Vihavainen. Predicting students' performance in an introductory programming course using data from students' own programming process. In IEEE 13th International Conference on Advanced Learning Technologies, pages 498–499, 2013.

[29] Watson, C., Li, F. W., & Godwin, J. L. (2014, March). No tests required: comparing traditional and dynamic predictors of programming success. In Proceedings of the 45th ACM technical symposium on Computer science education (pp. 469-474).

[30] West, M., Herman, G.L. and Zilles, C., 2015, June. Prairielearn: Mastery-based online problem solving with adaptive scoring and recommendations driven by machine learning. In 2015 ASEE Annual Conference & Exposition (pp. 26-1238). https://www.prairielearn.org/, accessed 2022.

[31] Wilson, B. C., & Shrock, S. (2001, February). Contributing to success in an introductory computer science course: a study of twelve factors. In ACM SIGCSE Bulletin (Vol. 33, No. 1, pp. 184-188). ACM.

[32] Zhang, Y., Fei, Q., Quddus, M., & Davis, C. (2014). An examination of the impact of early intervention on learning outcomes of at-risk students. Research in Higher Education Journal, 26.

[33] zyBooks.com, accessed 2022.