When to Check In?

Identifying Early Signs of Student Struggle at Various Cut-Points in CS1 Course Data

Presented by Abigail Liu

Motivation

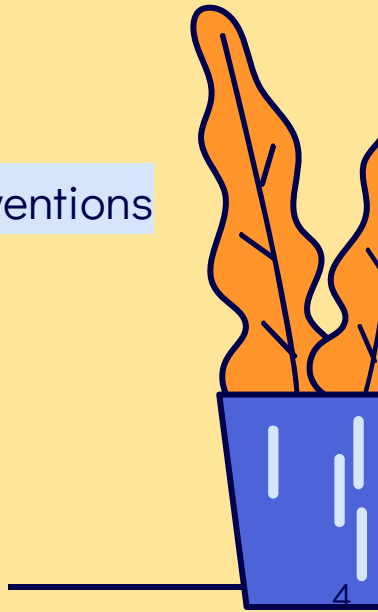
- According to the 2024 CRA Taulbee Survey, introductory computer science (CS1) courses have received
  - Large amounts of enrollment (Petersen 2016)
  - Sizable dropout rates (Petersen 2016, Tims 2023)

Motivation

- According to the 2024 CRA Taulbee Survey, introductory computer science (CS1) courses have received
  - Large amounts of enrollment (Petersen 2016)
  - Sizable dropout rates (Petersen 2016, Tims 2023)
- Students do not always ask for help on their own
  - (Hao 2017, Smith IV 2020, Kazerouni 2021)

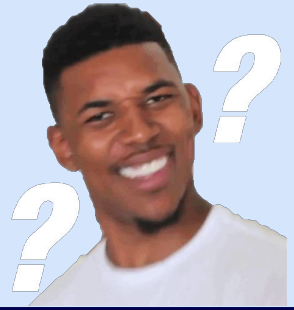
Motivation

- According to the 2024 CRA Taulbee Survey, introductory computer science (CS1) courses have received
  - Large amounts of enrollment (Petersen 2016)
  - Sizable dropout rates (Petersen 2016, Tims 2023)
- Students do not always ask for help on their own
  - (Hao 2017, Smith IV 2020, Kazerouni 2021)
- Both factors make it difficult for CS1 instructors to provide interventions that are timely and personalized to students

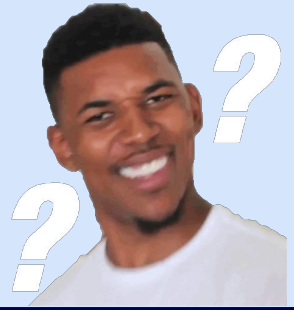


Our work aims to design technologies to help CS1 instructors, using learning analytics (LA) to predict student struggle and performance

Research Questions

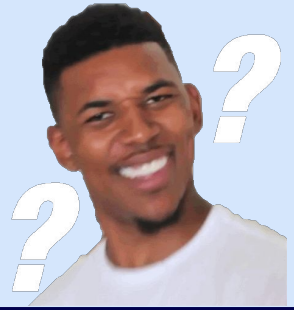


Research Questions



RQ1: Can we predict student performance using common behavioral data features from LA literature?

Research Questions



RQ1: Can we predict student performance using common behavioral data features from LA literature?

RQ2: Which data features are most predictive at different times in the semester?

Related Work

- Educational data mining uses action log data to predict student performance
  - Logs are a series of actions
  - often in version control systems (VCS) or learning management systems (LMS)

Related Work

- Educational data mining uses action log data to predict student performance
  - Logs are a series of actions
  - often in version control systems (VCS) or learning management systems (LMS)
- Arakawa et al. (2022)
  - Data from 312 students using GitHub
  - Logged each time a student pushed a version of code to their repository

Related Work

- Educational data mining uses action log data to predict student performance
  - Logs are a series of actions
  - often in version control systems (VCS) or learning management systems (LMS)
- Arakawa et al. (2022)
  - Data from 312 students using GitHub
  - Logged each time a student pushed a version of code to their repository
- Pinto et al. (2021)
  - Data from 733 students using PrairieLearn
  - Logged progress on programming homework, weekly quizzes, and cumulative exams

Current Datasets

- Small student samples
- Hard to verify claims

Current Features and Predictive Methods

- Varying sets of data features
- Hard to compare works

So how are we different from prior work?

Current Datasets

- Small student samples
- Hard to verify claims

Current Features and Predictive Methods

- Varying sets of data features
- Hard to compare works

Our Dataset

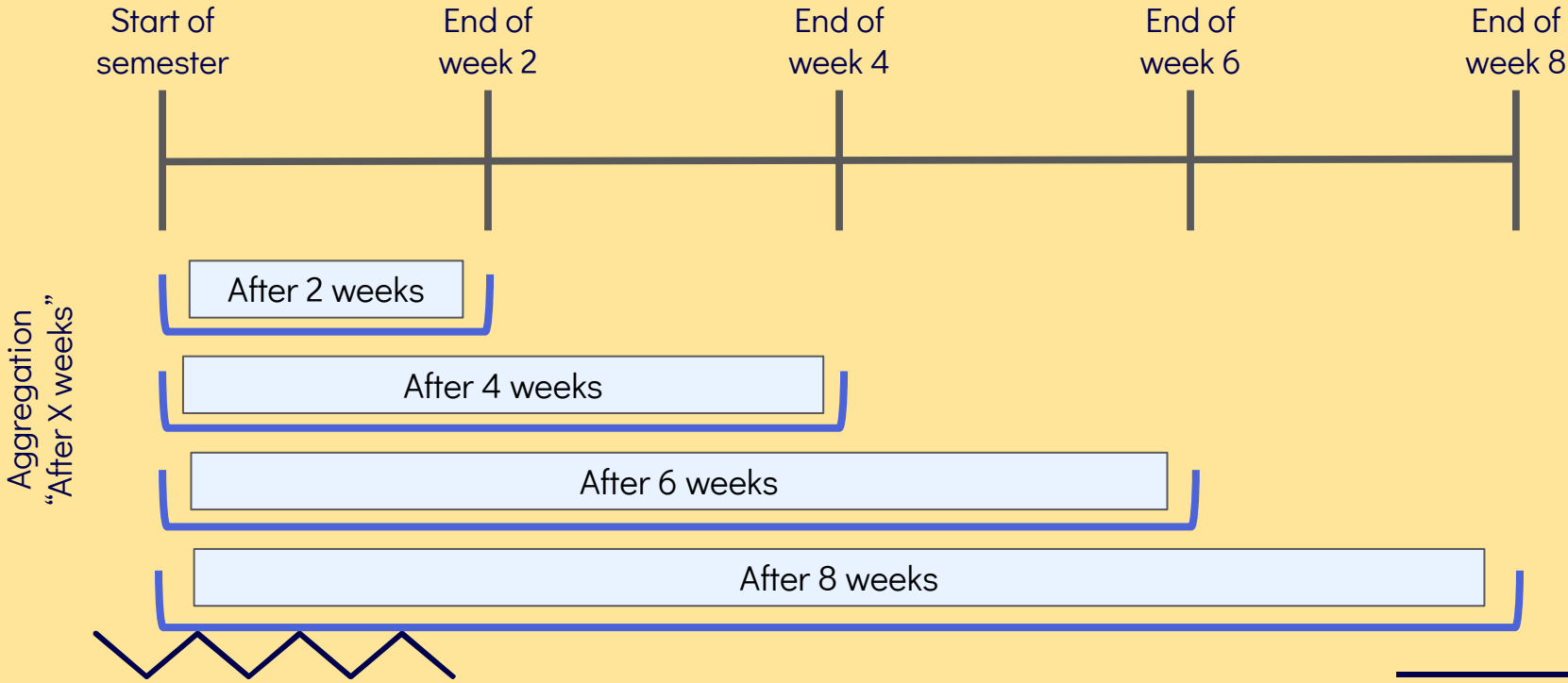
- Large student sample from 30 course sections
- Granular data collection allows for adaptable analyses

Our Features and Predictive Methods

- Compare 16 features on a single dataset
- Using a commonly used regression model

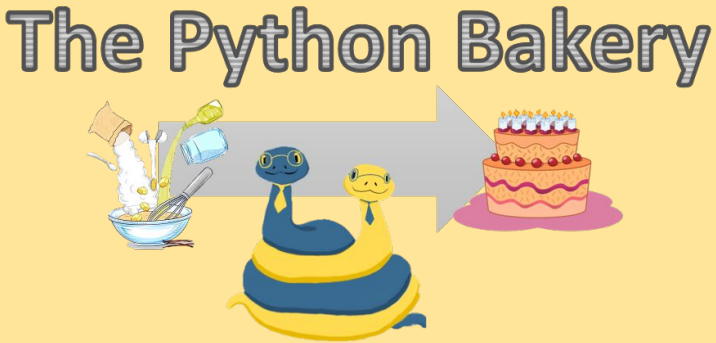
We examine how predictive LA data features are at different cut-points across the semester

Data Aggregation Methodology



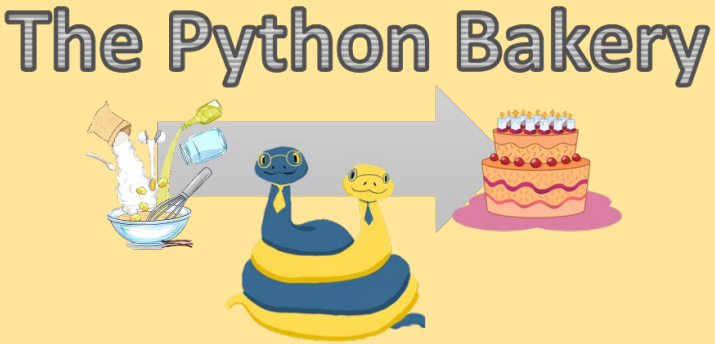
Bakery Evidence of Learning (EOL) Dataset

- 30 course sections between the Fall 2022 and Spring 2024 semesters
  - 21 sections for non-CS majors
  - 9 sections for CS majors



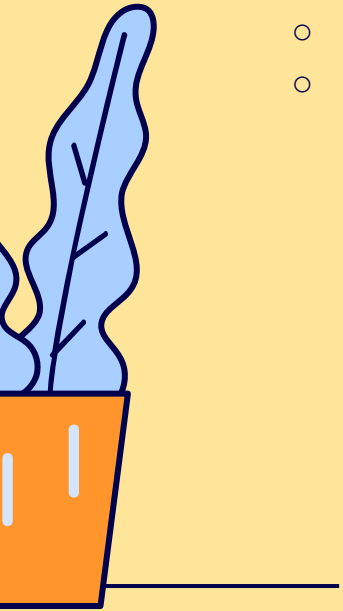
Bakery Evidence of Learning (EOL) Dataset

- 30 course sections between the Fall 2022 and Spring 2024 semesters
  - 21 sections for non-CS majors
  - 9 sections for CS majors
- Logged every time a student compiles code for an assignment
  - ~1,200 student samples
  - ~2.2 million student compiles/ assignment submissions
  - ~15 million unique code states



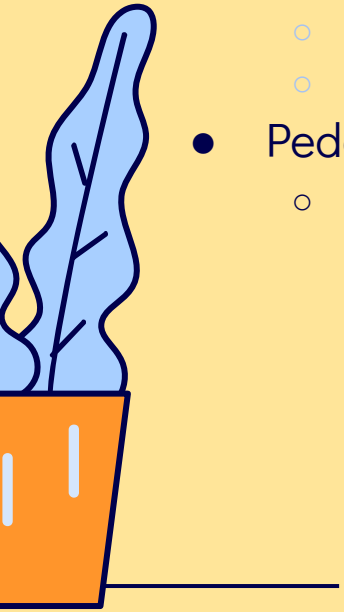
Dataset Analysis

- Focus on 14 coding assignments from Python Bakery Curriculum, which cover fundamental concepts like
  - Functions
  - If statements
  - Data structures



Dataset Analysis

- Focus on 14 coding assignments from Python Bakery Curriculum, which cover fundamental concepts like
  - Functions
  - If statements
  - Data structures
- Pedal is an autograding framework for analyzing student code
  - Gave feedback about unit tests, syntax errors, and other kinds of structural checks



Data Analysis

- We focus on coding assignments because they occur regularly throughout the semester

Data Analysis

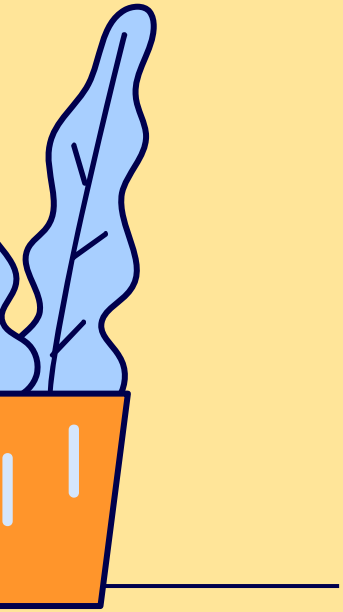
- We focus on coding assignments because they occur regularly throughout the semester
- Captures more nuance in student course progression

Data Analysis

- We focus on coding assignments because they occur regularly throughout the semester
- Captures more nuance in student course progression
- May also reveal signs of struggle much earlier than exams or projects.

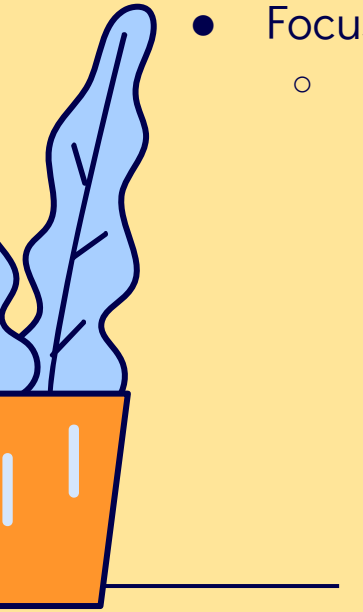
Model

- Generalized linear mixed models (GLMMs), used in prior research
  - Focused on logistic regression coefficients and p-values to assess the relationships between data features and course outcomes



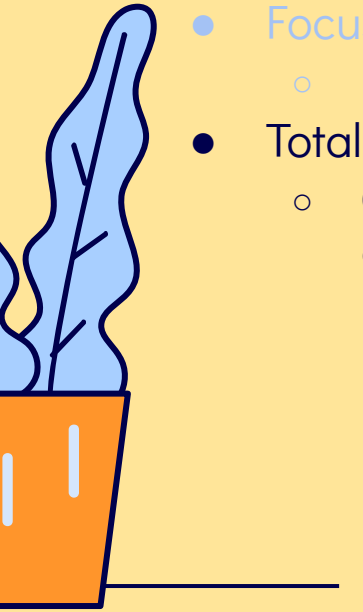
Model

- Generalized linear mixed models (GLMMs), used in prior research
  - Focused on logistic regression coefficients and p-values to assess the relationships between data features and course outcomes
- Focus on deviance scores to assess how well our models fit the data
  - Lower values indicate a betterfit.



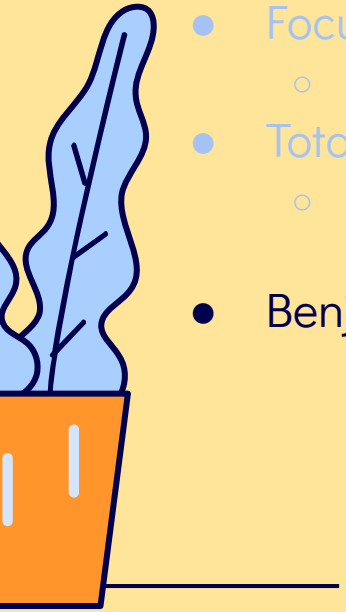
Model

- Generalized linear mixed models (GLMMs), used in prior research
  - Focused on logistic regression coefficients and p-values to assess the relationships between data features and course outcomes
- Focus on deviance scores to assess how well our models fit the data
  - Lower values indicate a betterfit.
- Total of 8 separate regressions
  - GLMM for 2, 4, 6, and 8 weeks of the semester for each of our two outcomes: final exam grades and final course grades

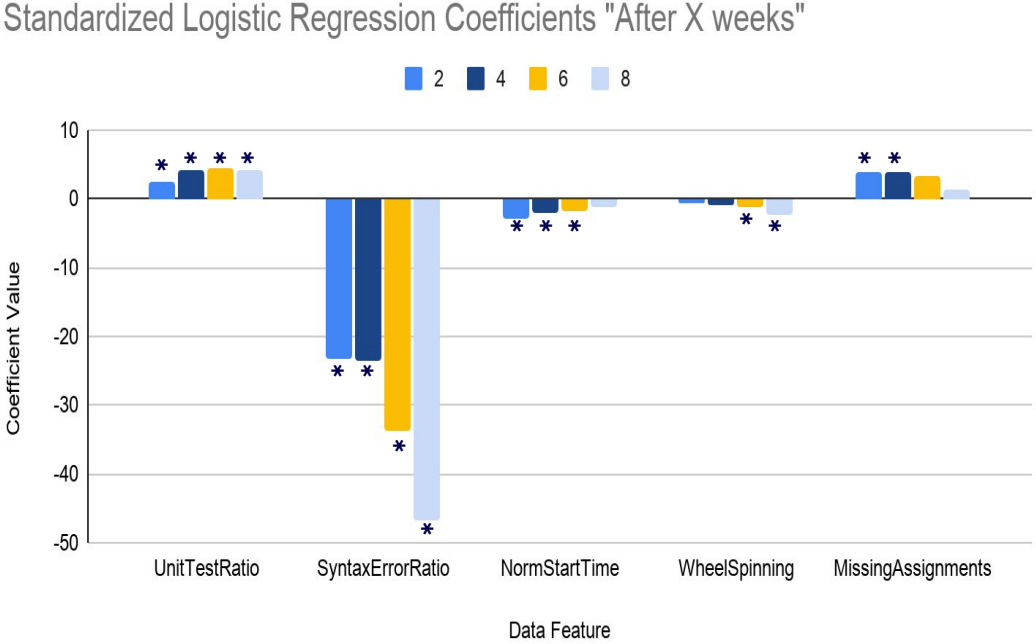


Model

- Generalized linear mixed models (GLMMs), used in prior research
  - Focused on logistic regression coefficients and p-values to assess the relationships between data features and outcomes
- Focus on deviance scores to assess how well our models fit the data
  - Lower values indicate a betterfit.
- Total of 8 separate regressions
  - GLMM for 2, 4, 6, and 8 weeks of the semester for each of our two outcomes: final exam grades and final course grades
- Benjamini-Hochberg procedure to account for the false discovery rate

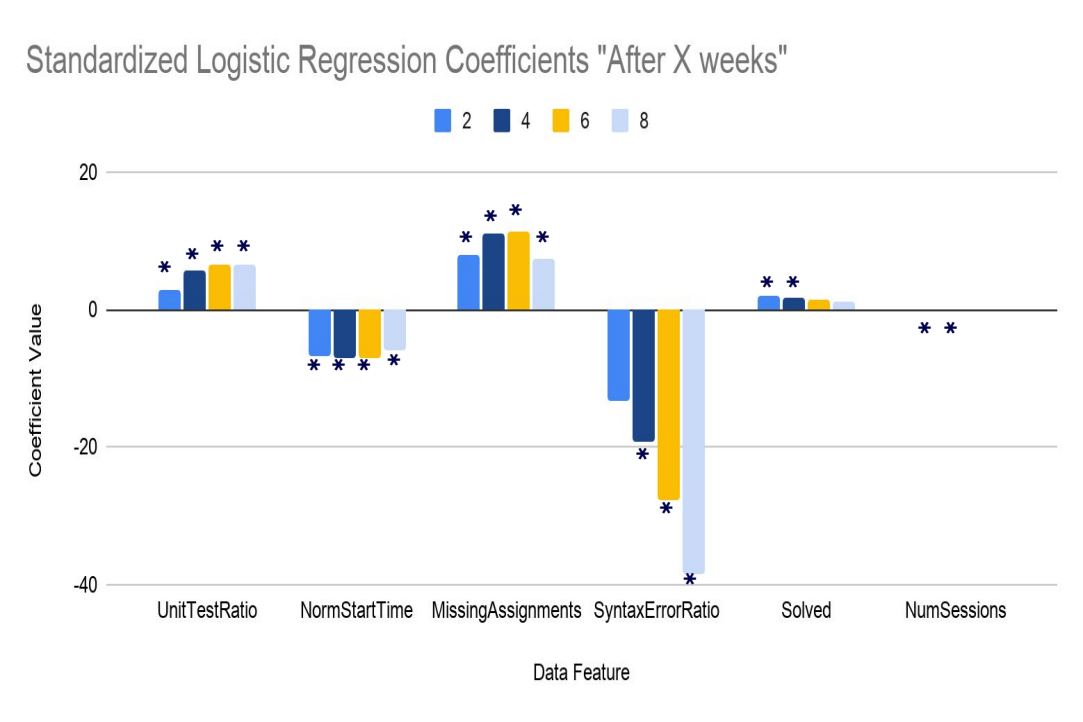


Results: Predicting Final Exam Grades



- Statistically significant across all cut-points
  - Accuracy in unit test problems
  - Amount of syntax errors
- Statistically significant at a subset of cut-points
  - How early a student starts an assignment
    - (after 2, 4, and 6 weeks)
  - Whether a student experiences wheel-spinning
    - (after weeks 6 and 8)
  - How many missing assignments
    - (after 2 and 4 weeks)
- Decreasing deviance scores →
  - Model fits better with more semester data

Results: Predicting Final Course Grades



- Statistically significant across all cut-points
  - Accuracy in unit test problems
  - How early a student starts an assignment
  - How many missing assignments
- Statistically significant at a subset of cut-points
  - Amount of syntax errors
    - (after weeks 4, 6, and 8)
  - Whether a student got 100% on an assignment
    - (after weeks 2 and 4)
  - Number of coding sessions
    - (after weeks 2 and 4)
- Decreasing deviance scores →
  - Model fits better with more semester data
  - Model fits better with final course grade

Discussion + Future Work

1. During the semester, the complexity of course materials increases
   a. Students' coding behaviors may also change across the semester
   b. These changes are reflected in the statistical significance for some data features

Discussion + Future Work

1. During the semester, the complexity of course materials increases
   a. Students' coding behaviors may also change across the semester
   b. These changes are reflected in the statistical significance for some data features
2. Students may attempt fewer assignments later in the semester
   a. Due to several factors, like balancing effort between other courses or losing motivation
   b. This decrease in attempts affects effort-based data features

Discussion + Future Work

1. During the semester, the complexity of course materials increases
   a. Students' coding behaviors may also change across the semester
   b. These changes are reflected in the statistical significance for some data features
2. Students may attempt fewer assignments later in the semester
   a. Due to several factors, like balancing effort between other courses or losing motivation
   b. This decrease in attempts affects effort-based data features
3. Our chosen data features do not require any special calculations
   a. Could be implemented into a dashboard for instructors
   b. Could be used to identify patterns and early warning signs within student behavior, allowing instructors to intervene in a timely manner

Limitations

- Analysis did not include other course materials like
  - Midterm exams
  - Projects
  - More advanced assignments (in 2nd half of semester)

Limitations

- Analysis did not include other course materials like
  - Midterm exams
  - Projects
  - More advanced assignments (in 2nd half of semester)
- We only test at cut-points at 2-week intervals

Limitations

- Analysis did not include other course materials like
  - Midterm exams
  - Projects
  - More advanced assignments (in 2nd half of semester)
- We only test at cut-points at 2-week intervals
- Our dataset is from one institution and only 2 CS1 courses

Conclusion

- Compared 16 common behavioral data features from LA literature to predict student struggle and performance

Conclusion

- Compared 16 common behavioral data features from LA literature to predict student struggle and performance
- For the final exam and course grade, student accuracy on unit test problems was a statistically significant predictor across all cut-points
    - Statistical significance of other features changed as the semester progressed

Conclusion

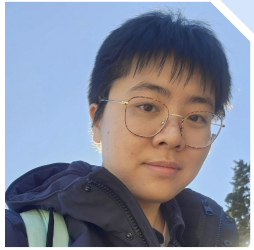
- Compared 16 common behavioral data features from LA literature to predict student struggle and performance
- For the final exam and course grade, student accuracy on unit test problems was a statistically significant predictor across all cut-points
  - Statistical significance of other features changed as the semester progressed
- These data features can be implemented into a dashboard tool to help CS1 instructors provide timely student interventions, powered by temporally contextual analysis

Special thanks to my collaborators!

Contact Info:

apliu@udel.edu

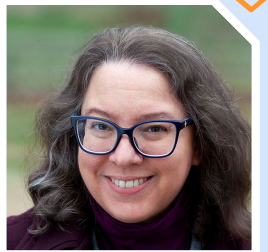
www.linkedin.com/in/abigailpliu/



Abigail Liu

Sammy Alashoush

Austin Cory Bart



Teomara Rutherford

Check out our paper!

Nazim Karaca

John Aromando Jr.

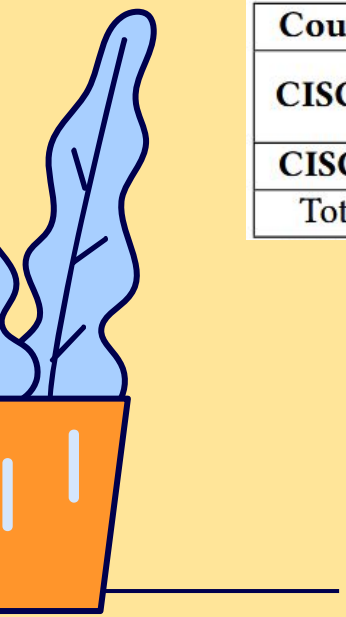
Matthew Mauriello

MADRID2026 ITiCSE

SENSIFY LAB

EOL Dataset Information

Course	Number of Sections	Number of Students	Description
CISC-X	21 (15 non-honors)	853 (730 non-honors)	For non-CS major students in the College of Engineering
CISC-Y	9 (8 non-honors)	336 (306 non-honors)	For CS majors
Total	30	1,189	



EOL Data Information

	FinalExamGrade	FinalCourseGrade
≥80	881	868
<80	301	314
Assignments	**Week Available**	**# of Problems**
Intro A/B	1	25/39
Functions A/B	2	28/21
If A/B	3	6/15
Structures A/B	4	21/21
For A/B	6	14/12
Sequences A/B	7	12/11
Nesting A/B	8	11/9

Data Features Literature Review

Meta-Category	Description
Student Info	Course/Demographic information about students [22]
Student Prep	Students' prior knowledge and skills [19]
Achieving Mastery	Accuracy measures indicating that a student is learning. Includes unit tests, grades, correct/incorrect submissions, if the student eventually got the correct answer [2, 4, 5, 15, 18, 22, 26, 28, 39]
Code State Contents	Contents of a particular code state, not looking at its relationship to other code states [2, 5, 21]
Course/System Info	Information about the course or learning system [22, 26, 39]
Changes in Code States	What/how much a student is changing in their code [4, 15, 21, 37]
Difficulty of Task	Related to how challenging the assignment is. Includes student perceptions of how hard a piece of code is to implement [2]
Effort	Related to how many attempts a student used to complete an assignment [2, 4, 5, 8, 13, 15, 22, 26, 28, 37, 39]
Engagement	Related to how active a student is in completing an assignment, usually about how often/consistent students work [5, 15, 21, 28, 37]
Handling Errors	Related to resolving errors in code (e.g., syntax and runtime errors) [4, 5, 15]
Procrastinating	Related to when a student works on an assignment [4, 5]
Struggle	Indicators of students getting stuck. Includes wheel-spinning [2, 4, 28, 37]
Time-on-Task	Related to the amount of time a student works on an assignment [2, 5, 15, 18, 22, 28, 28]

Table 1: Meta-categories for data features. The descriptions and works that use features from each are listed.

Features used in this work

Meta-Categories	Feature Name	Description	Range
Student Info	Semester	Which semester a student's course was in (0 for fall, 1 for spring) [22]	[0,1]
Student Info	IsHonors	If a student was in a non-Honors (0) or Honors (1) course section [22]	[0,1]
Student Info	IsCSMajor	If a student was a non-CS major (0) or CS major (1) [22]	[0,1]
Student Prep	PretestScore	Score student received for the pretest to the course (4 indicates 100%) [19]	[0,4]
Achieving Mastery	Solved*	If a student got 100% on an Assignment before the lock date. (0 if no, 1 if yes) [2, 28]	[0,1]
Achieving Mastery	UnitTestRatio*	[# of passed unit tests]/[total # of unit tests] in an Assignment [4, 39]	[0,1]
Effort	NumSubmissions	[# of submissions]/[total # of problems after X weeks] [2, 5, 8, 13, 22, 26, 28, 39]	[0, ∞]
Engagement, Effort	NumSessions	[# of coding sessions]/[# of Assignments after X weeks] A coding session ends when no action is done after 15 minutes [28]	[0, ∞]
Engagement, Effort	Missing Assignments	[# of unattempted Assignments] /[# of Assignments after X weeks] [15]	[0, 1]
Handling Errors, Effort	SyntaxErrorRatio*	[# of syntax error messages]/[total # of feedback messages] in an Assignment [4, 5]	[0, 1]
Procrastinating	NormStartTime*	Normalized timestamp of the first submission of an Assignment. 0, 0.5, and 1 are when an Assignment is available, due, and locked, respectively [4]	[0, 1]
Struggle	NumRapidGuesses	[# of rapid guesses]/[# of Assignments after X weeks] A rapid guess is when ≥ 3 submissions are made in a row in < 15 seconds [28]	[0, ∞]
Struggle	WheelSpinning*	If a student attempted the Assignment but never got 100%, or got 100% with more submissions than 75% of other students. (0 if no, 1 if yes) [2]	[0, 1]
Struggle	NumExcessRuns	[# of times code was compiled multiple times without any changes] /[# of Assignments after X weeks] [37]	[0, ∞]
Struggle	NumFeedback Unchanged	[# of submissions where the feedback message did not change] /[# of Assignments after X weeks] [4]	[0, ∞]
Time-on-Task	TotalTime SpentCoding	[# seconds spent coding]/[# of Assignments after X weeks] from latencies between keystrokes excluding breaks over 15 minutes [18]	[0, ∞]

Table 2: Data features used in this work. The description, citations using similar metrics, and range are provided. * indicates that a feature was aggregated using weighted average. Weights were determined by the number of problems in each assignment.

Results: Predicting Final Exam Grades

	Standardized Logistic Regression Coefficients "After X weeks"			
Data Feature	2	4	6	8
Semester	-0.27*	-0.24*	-0.24*	-0.23
IsHonors	1.38*	1.41*	1.42*	1.41*
IsCSMajor	0.32*	0.30*	0.30*	0.29*
PretestScore	0.08	0.08	0.07	0.11
Solved	0.55	-0.06	-0.45	-0.92
UnitTestRatio	2.56*	4.14*	4.56*	4.17*
NumSubmissions	>-0.01	0.03	0.05	0.06
NumSessions	>-0.001	>-0.001	>-0.001	>-0.001
SyntaxErrorRatio	-23.40*	-23.61*	-33.70*	-46.69*
NormStartTime	-2.79*	-2.12*	-1.86	-1.30
NumRapidGuesses	-0.01	>-0.01	-0.01	-0.02
WheelSpinning	-0.63	-0.870	-1.31	-2.26*
NumExcessRuns	-0.02*	-0.03	-0.02	-0.04
NumFeedback Unchanged	<0.001*	<0.001	<0.001	<0.001
TotalTime SpentCoding	<0.001*	<0.001	<0.001	<0.001
MissingAssignments	3.99*	3.97*	3.27	1.27
Deviance Score	1075.70	1038.32	1023.34	1003.68

Table 3: Standardized logistic regression coefficients, for the final exam grade outcome. * indicates *p*-value <.05 after applying the Benjamini-Hochberg procedure.

- Statistically significant across all cut-points
  - Accuracy in unit test problems
  - Amount of syntax errors
- Statistically significant at a subset of cut-points
  - How early a student starts an assignment
    - (after 2, 4, and 6 weeks)
  - Whether a student experiences wheel-spinning
    - (after weeks 6 and 8)
  - How many missing assignments
    - (after 2 and 4 weeks)
- Decreasing deviance scores →
  - Model fits better with more semester data

Results: Predicting Final Course Grades

	Standardized Logistic Regression Coefficients "After X weeks"			
Data Feature	2	4	6	8
Semester	-0.46*	-0.55*	-0.58*	-0.50*
IsHonors	2.16*	2.28*	2.27*	2.12*
IsCSMajor	0.62*	0.74*	0.77*	0.73*
PretestScore	0.04	0.02	0.02	0.04
Solved	1.93*	1.79*	1.51	1.11
UnitTestRatio	2.78*	5.62*	6.47*	6.45*
NumSubmissions	0.02	0.08*	0.11*	0.12*
NumSessions	>-0.001*	>-0.001*	>-0.001	>-0.001
SyntaxErrorRatio	-13.16	-19.19*	-27.80*	-38.57*
NormStartTime	-6.89*	-7.08*	-7.00*	-5.86*
NumRapidGuesses	-0.01	-0.01	-0.02	-0.02
WheelSpinning	-0.25	-0.24	-0.81	-1.74
NumExcessRuns	-0.01	-0.02	-0.02	-0.04
NumFeedback Unchanged	<0.001	>-0.001	>-0.001	>-0.001
TotalTime SpentCoding	<0.001	<0.001	<0.001	<0.001
MissingAssignments	8.04*	10.93*	11.40*	7.42*
Deviance Score	978.42	880.68	854.56	817.04

Table 4: Standardized logistic regression coefficients, for the final course grade outcome. * indicates *p*-value <.05 after applying the Benjamini-Hochberg procedure.

- Statistically significant across all cut-points
  - Accuracy in unit test problems
  - How early a student starts an assignment
  - How many missing assignments
- Statistically significant at a subset of cut-points
  - Amount of syntax errors
    - (after weeks 4, 6, and 8)
  - Whether a student got 100% on an assignment
    - (after weeks 2 and 4)
  - Number of coding sessions
    - (after weeks 2 and 4)
- Decreasing deviance scores →
  - Model fits better with more semester data
  - Model fits better with final course grade

Deviance

More precisely, the deviance is defined as the difference of likelihoods between the fitted model and the saturated model:

$$D = -2 \log \text{lik}(\hat{\boldsymbol{\beta}}) + 2 \log \text{lik}(\text{saturated model}).$$

Since the likelihood of the saturated model is exactly one[31], then the deviance is simply another expression of the likelihood:

$$D = -2 \log \text{lik}(\hat{\boldsymbol{\beta}}).$$

As a consequence, **the deviance is always larger or equal than zero, being zero only if the fit is perfect**.

GLMM

Generalized linear mixed models (or GLMMs) are an extension of linear mixed models to allow response variables from different distributions, such as binary responses. Alternatively, you could think of GLMMs as an extension of generalized linear models (e.g., logistic regression) to include both fixed and random effects (hence mixed models). The general form of the model (in matrix notation) is:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}$$

Where $\mathbf{y}$ is a $N \times 1$ column vector, the outcome variable; $\mathbf{X}$ is a $N \times p$ matrix of the p predictor variables; $\boldsymbol{\beta}$ is a $p \times 1$ column vector of the fixed-effects regression coefficients (the "betas"); $\mathbf{Z}$ is the $N \times q$ design matrix for the q random effects (the random complement to the fixed $\mathbf{X}$); $\boldsymbol{\gamma}$ is a $q \times 1$ vector of the random effects (the random complement to the fixed $\boldsymbol{\beta}$); and $\boldsymbol{\varepsilon}$ is a $N \times 1$ column vector of the residuals, that part of y that is not explained by the model, $\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma}$. To recap:

$$\overbrace{\mathbf{y}}^{N\times 1} = \overbrace{\underbrace{\mathbf{X}}_{N\times p}\ \underbrace{\boldsymbol{\beta}}_{p\times 1}}^{N\times 1} + \overbrace{\underbrace{\mathbf{Z}}_{N\times q}\ \underbrace{\boldsymbol{\gamma}}_{q\times 1}}^{N\times 1} + \overbrace{\boldsymbol{\varepsilon}}^{N\times 1}$$

Thank you!

When to Check In?

Identifying Early Signs of Student Struggle at Various Cut-Points in CS1 Course Data

Presented by Abigail Liu
ITiCSE 2026

Abigail Liu, Sammy Alashoush, Austin Cory Bart, Teomara Rutherford, Nazim Karaca, John Aromando Jr., and Matthew Louis Mauriello. 2026. When to Check In?: Identifying Early Signs of Student Struggle at Various Cut-Points in CS1 Course Data. In Proceedings of the 31st ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2026), July 10–15, 2026, Madrid, Spain. ACM, New York, NY, USA, 7 pages.https://doi.org/10.1145/3803400.3809327

References

- Kai Arakawa, Qiang Hao, Wesley Deneke, Indie Cowan, Steven Wolfman, and Abigayle Peterson. 2022. Early identification of student struggles at the topic level using context-agnostic features. In Proceedings of the 53rd ACM Technical Symposium on Computer Science Education-Volume 1. 147–153.
- Augie Doebling and Ayaan M Kazerouni. 2021. Patterns of academic help-seeking in undergraduate computing students. In Proceedings of the 21st Koli Calling International Conference on Computing Education Research. 1–10.
- Qiang Hao, Brad Barnes, Robert Maribe Branch, and Ewan Wright. 2017. Predicting computer science students' online help-seeking tendencies. Knowledge Management & E-Learning: An International Journal 9, 1 (2017), 19.
- Andrew Petersen, Michelle Craig, Jennifer Campbell, and Anya Tafliovich. 2016. Revisiting why students drop CS1. In Proceedings of the 16th Koli Calling International Conference on Computing Education Research. 71–80.
- Juan D Pinto, Yingbin Zhang, Luc Paquette, and Aysa Xuemo Fan. 2021. Investigating elements of student persistence in an introductory computer science course. In 5th Educational Data Mining in Computer Science Education (CSEDM) Workshop.
- Katie Siek and Jasmine Batten. 2024. Taulbee Survey 2024: Annual Report. Computing Research Association (2024).
- David H Smith IV, Qiang Hao, Vanessa Dennen, Michail Tsikerdekis, Bradly Barnes, Lilu Martin, and Nathan Tresham. 2020. Towards Understanding Online Question & Answer Interactions and their effects on student performance in large-scale STEM classes. International Journal of Educational Technology in Higher Education 17 (2020), 1–15.
- Jodi L Tims, Cindy Tucker, Mark A Weiss, and Stuart Zweben. 2023. Computing Enrollment and Retention: Results from the 2021–22 Undergraduate Enrollment Cohort. ACM Inroads 14, 4 (2023), 24–43.