Using Course Evaluations and Student Data to Predict Computer Science Student Success

Anlan Du*
amd5wf@virginia.edu
University of Virginia
Charlottesville, VA, United States

Alexandra Plukis*
aplukis@asu.edu
Arizona State University
Tempe, AZ, United States

Huzefa Rangwala
hrangwal@gmu.edu
George Mason University
Fairfax, VA, United States

ABSTRACT

As the field of computer science has grown, the question of how to improve retention in computer science, especially for females and minorities, has grown increasingly important. Previous research has looked into attitudes among those who leave CS, as well as the impact of taking specific courses; we build on this body of research using large-scale analysis of course evaluations and students' academic history. Our goal is to understand their potential connection to a student's performance and retention within the CS major. We process course-specific data, faculty evaluations, and student demographic data through various machine learning-based classifiers to understand the predictive power of each feature. We find our algorithm performs significantly better for higher-performing students than lower-performing ones, but do not find that evaluations significantly improve predictions of students doing well in courses and staying in the major.

KEYWORDS

educational data mining, course evaluations, computer science, retention, grade prediction, algorithm fairness

1 INTRODUCTION

Among the most important aspects of a college education are the classes a student takes. Often, college students use introductory courses to decide what they would like to study and pursue. Bad experiences in an introductory course might detract from a student's first impression of a field, while a good experience in a course might improve his or her opinion, even boosting retention and improving skills upon graduation [13]. Therefore, it is key that administrators and professors alike understand which course characteristics maintain interest and improve student outcomes. Such information can impact administrative decisions, such as who is assigned to teach particular courses and the recommended sequence of courses.

The digitization of student records and course evaluations offers a unique opportunity to apply big data modeling techniques to study retention. George Mason University, the data source for this work, keeps anonymized records on students' academic records in high school, demographic data, and their course loads and grades at the university. They also administer standardized course evaluations across all courses. Various data mining and modeling techniques, such as decision trees and support vector machines, can be applied to these datasets and their results compared. Using this data, one can more easily find patterns that reveal how different traits affect student retention.

George Mason also offers a unique opportunity to analyze the impact of professor gender on student success. George Mason's engineering faculty is 26.8% female, more than 1.5 times higher than the national average of 15.7% [10][16]. A larger female faculty means that analyses of the impact of instructor gender are less likely to be swayed by a single professor and therefore more statistically significant.

2 RELATED WORK

Our work builds upon previous research regarding student both college retention and achievement in courses, both generally and between demographic groups [7] . Demographic disparities are particularly evident in the number of degrees awarded. For instance, during George Mason University's 2017-2018 school year only 15.8% of the total 196 computer science (CS) degrees were awarded to females. This lack of representation is even more pronounced for minority students—only 6 CS degrees were awarded to African American students and 16 awarded to their Hispanic counterparts [9]. These disparities have led to a large body of research into retention for minorities in STEM and specifically [8][1][15]. Bettinger and Long researched the impact of female faculty on female retention in majors or repeated interest in classes and found mixed results: some disciplines such as statistics and mathematics benefited from an early female professor introduction, while others saw a decrease in female retention. The authors pointed out that it was difficult to gauge the exact impact of female professors in fields that had low levels of females in faculty, such as engineering and physics. We hope to improve upon on this because George Mason's School of Engineering female full time academic faculty make up 26.8%, far surpassing the national average of 15.7% [16] [10].

The issue of student performance and retention extends beyond under-represented minorities. Cucuringu et al. used fifteen years of student data to find classes that optimized a student's likelihood of successfully completing a course of study with high grades [5]. They also took the step of segmenting a student population into sub-groups based on various characteristics, so as to understand the nuances that different types of students might experience. Morsy and Karypis used a similarly broad, qualitative approach to predict student performance based on previous classes taken [14].

Research specific to CS retention has also been conducted: Biggers et al. incorporated interviews of students who left CS, seeking to find the qualitative sentiments that affected both female and male students' decisions [2]. We combine these two approaches by using data on students' individual demographics, grades, and course history to understand how each factor may contribute to

*Both authors contributed equally to this research.

both student performance and choice of major. Additionally, we incorporate student evaluations for the courses they take to understand the role that these qualitative elements may play in these outcomes, as suggested by Biggers et al [2].

Research on course evaluations suggests they may prove informative with regards to a student's academic experience. Much research has studied the relationship between the ease of a course, often represented by the grade a student receives, and the rating of the faculty. One well-known meta-analysis by Cohen argued that students are fairly accurate in their assessments of instructional efficacy [6]. Centra's study built upon this notion, and further emphasized that students do not give higher evaluations to professors in a quid pro quo for higher grades: both extremely easy and difficult courses suffered in student evaluations, while courses with appropriate difficulty received the best evaluations [3]. Feldman analyzed the contributory power of various teacher characteristics to a teacher's overall rating and student achievement, finding that preparation, organization, clarity, and students' feelings of engagement contributed most strongly to overall performance [11]. He also highlighted some myths about student evaluations, citing research that suggests that they can, in fact, be informative. We incorporate evaluations in order to expand on these questions of student evaluation efficacy, and understand what they say about students' experiences and choices.

3 PROBLEM DESCRIPTION

The objective of this study is to investigate a few questions relating course quality—defined using faculty traits such as gender and instructional evaluations—to student retention in computer science. Specifically, we will address the following inquiries:

(1) Which course features, if any, in lower division CS courses improve graduation retention for students?
(2) Which course features, if any, of instructors in introductory CS courses can predict student success in future CS courses?
(3) Do non-CS courses that are required by CS majors, like calculus, have an impact on major retention for students? If so, which courses and features have the largest impact?

4 MATERIALS

4.1 Dataset

Our dataset consisted of records containing first time freshman student enrollment and course evaluation data for 20,825 George Mason students over the span of eight years, from Summer 2009 to Fall 2018. All student data were collected and anonymized in accordance with GMU's Institutional Review Board policies. The student data contained demographics data such as age, sex, and race; admissions data such as high school, SAT score, and high school GPA; and course data such as declared major, graduation year, courses taken, and grades received. Students who transferred into GMU were not included in this dataset because they likely had completed introductory courses at their previous institutions, rendering that first-year data inaccessible to us. We also collected course evaluation data on 87,629 GMU courses from Summer 2009 to Spring 2019, 8,243 of which were computer science, or computer science-adjacent courses. The evaluations are averages of all of the student evaluations for that specific course and section, so there is 1 evaluation available for each unique course GMU offers. This data was collected from the GMU evaluation site [1], which is publicly available while on campus. As these are publicly available documents on campus and the identifying features were anonymized, they are exempt research under GMU's IRB policy [2]. To collect data on professor gender, we reviewed pronoun usage in departmental documents and consulted faculty members when documentation was insufficient.

The courses we describe as CS-adjacent are courses taught by or in conjunction with the Department of Computer Science at GMU. These CS-adjacent courses include Information Technology, Computer Game Design, Software Engineering, Electrical and Computer Engineering, and Information Systems. After discarding course data with no grades or grades not translating to the A-F scale and applying our course filters, we had records for 57,627 student-course enrollments.

4.2 Definitions

We frequently discuss "student success" within the computer science major. In this paper, our definition of "success" is divided into three categories:

Completion of a computer science degree: A student is defined as graduating with a computer science degree if he or she graduated with a major in either computer science or applied computer science. A student is defined as not graduating with a CS degree if he or she graduated, but not with a CS major. Because we are focused on retention, not graduation, we only included in our data students who had had enough time to graduate. By not including students who transfer or drop out of GMU, or who simply have not graduated yet, we reduced the number of confounding variables that are not directly related to students' experiences in CS.

Fulfilment of a student's potential in a course: A student's "potential" in CS211 is defined as the term GPA of the semester in which CS112—the direct pre-requisite—was taken. Our interest in this stems from its potential in combination with predictions of passing a course. Students who perform below their "potential" within CS211, despite passing and receiving credit for the course, might still benefit from administrator involvement. Alternatively, the characteristics of students performing above their potential may highlight positive factors that should continue to be proliferated on an institutional level.

Passing a course for credit: A student is defined as passing a course for credit if he or she receives a C grade or above. At GMU, computer science BS students "must earn a C or better in any course intended to satisfy a prerequisite for a computer science course ... [s]tudents may attempt an undergraduate course taught by the Volgenau School of Engineering twice." [3]. In our research, we specifically target student success in CS112 and CS211 because they are required courses for CS/ACS majors and pre-requisites for all other programming courses. Figure 1 visualizes the contrast in pass rates for first and second attempts in CS211: within our dataset,

[1] https://irr2.gmu.edu/
[2] https://rdia.gmu.edu/topics-of-interest/human-or-animal-subjects/human-subjects/exempt-research/
[3] https://catalog.gmu.edu/colleges-schools/engineering/computer-science/computer-science-bs/#admissionspoliciestext

only 19.8% of students attempting CS211 for the first time did not receive credit, versus 63.3% of students on their second attempt.

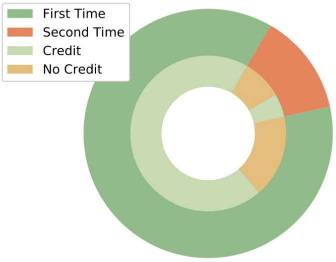

Figure 1: Receiving Credit for CS211.

5 METHODS

For this work, we compared performance of predictive models that were trained on three different sets of data, which are fully described in Appendix 9.2:

(1) Baseline predictions based on high school performance and student demographic data, as well as basic course information such as the term in which a course was taken and a student's GPA in that term.
(2) Baseline features in addition to instructor gender and course evaluations for the classes, either CS-only or math and CS, taken by each student.
(3) Baseline features, plus the course numbers as unique identifiers that were distinct for each section and semester of a class, but common to all of the students who took that section.

We chose to use machine learning classifiers because they can often pick up on more intricate patterns and correlations than linear and other basic statistical models can would. We decided to test these distinct data sets because they each highlight a component of student courses that may be significant to students' performance and ultimate retention. The full list of features used in each experiment are described in Appendix 9.

We used seven classifiers from the Python sci-kit learn library: Random Forest, Gradient Boosting, AdaBoost, SVC, Decision Tree, Neural Net, and Naive Bayes. For each of these models, we performed 5-fold cross-validation, recording the resulting the averages and standard deviations. In order to account for imbalances in our dataset, we decided upon area under an ROC curve (ROC AUC) and F1 score as our main metrics, because they take into account precision and recall in addition to overall accuracy.

5.1 Pre-Processing

We consolidated student data for all students who took at least one CS class, of whom there were 15,552. To better incorporate summer student data, we moved summer courses to the proceeding fall term. Then, we calculated percentile values for students' SAT scores and high school GPAs, enabling us to compare these metrics along a standard scale of 0 to 1. Next, for models predicting retention, we removed all students who had not yet graduated, leaving us with 7,602 students. Lastly, we dropped all students with empty values for any of the columns used in training. This left a dataset of 1,476 students who took at least one CS or CS-adjacent class before graduating. Of those, 330 graduated with a CS or ACS major, or 22.35%. This left us with an imbalanced dataset, leading to our decision to use F1 score and ROC AUC to characterize our models.

For the grade prediction portion, students who received no grade—meaning they audited or did not complete the class—were not included in the data. This left 1,728 students who took both CS112 and CS211 at GMU at least once. In the cases where students took these courses multiple times, only the initial course attempt was used so as to only capture their original experience in the class. Predicting grades for only first attempts of CS211 offers an earlier flagging system for at-risk students.

We wanted to understand the impact that not only general instructor qualities, but also "exemplary" instructors, had on student grades. To that end, each grade prediction model was run with the course evaluations processed in one of two ways: percentiles or flags. Percentiles, which capture the general quality of an instructor, had each evaluation entry into a percentile relative to the other courses. Flags, which served to identify exemplary instructors, transformed each entry into a binary feature based on whether it was in the top 10% of evaluation scores in that category.

Although evaluations offer more data than can usually be gleaned from student records, we tried to capture the elements in a course that cannot be captured in evaluations or records. We did so by creating unique course IDs for each course, so as to highlight especially good courses, good times of day for students, and good connections between students in courses—all of which are not explicitly quantified in our data.

5.2 Experiments

As mentioned previously, we had three main groups of datasets. The second group, which includes the course evaluation data, was then run on three different subsets: first, it was trained with just the "overall teaching" and "overall course" evaluation scores for the first CS and math courses, then the overall evaluations from the first two courses, then all available course evaluation metrics for the first two courses in each area. For graduation prediction, both math and CS courses were included in the evaluation data in order to capture a full snapshot of introductory courses. For grade prediction, only CS courses are included so as to not diminish the dataset of non-CS or non-STEM students, who often do not have the same rigorous math requirements.

Our rationale in deploying some tests with just two course evaluation features per course was that the added dimensionality of running the models on all of the features (many of which were positively correlated) might hinder performance. The baseline was meant to be the control for the predictive capabilities of only basic course features and student demographic information, so that subsequent tests might reveal how much predictive power the additional data might have added. The full list of features used for each of these experiments are listed in Appendix 9.1.

All of our experiments deal with binary classification, and as such require binary flagging for the classes of interest. In grade prediction experiments, those who are at risk—of either not receiving credit or not fulfilling their potential in a course—are flagged with a 1. In the graduation predictions, students who graduate with a computer science or applied computer science degree are flagged with a 1.

For each experiment, we ran 5-fold cross validation on our models, using a deterministic seed to generate our training-testing splits so that we could directly compare splits before and after the models were trained. We performed Student's t-tests on our results to understand the significance of any differences in performance.

5.3 Fairness

In order to check that the predictions were not favoring certain students already predisposed to graduating with a CS degree or passing their courses, we decided to separate the students into groups based on their academic abilities coming into college. We consider a prediction algorithm to be fair if its F1 score remains statistically similar regardless of the student's quartile standing. We used high school GPA (HS GPA) and total SAT scores to have one metric of school success and one metric of testing success to create a fuller understanding of student academic ability. These two scores were transformed into percentiles, averaged together, then transformed into a percentile once more. This final percentile calculation divided the students into evenly sized groups.

The students were then separated into 4 groups based on their percentile standings, as pictured in Figures 3 and 4. To test the fairness implications, 5-fold splits were trained on all students and then tested only on certain quartiles. This way, we could clearly see any disparity in performance for all students versus those in separate groups of students.

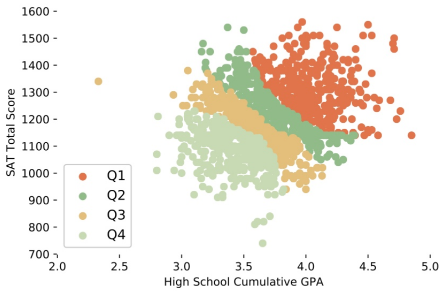

Figure 2: High School GPA versus SAT Total score for all non-transfer students who took both CS112 and CS211 at GMU.

We used these quartiles to test for fairness by training each of our models on the full datasets, splitting up the testing sets based on the quartiles, and calculating the metrics based on these results. We then compared these quartile results with the results for all students to determine if there was a significant difference between them, and therefore a disparity in fairness for differing groups.

6 RESULTS

Our results are divided into three sections:

(1) Performance metrics (F1 Score, ROC AUC, Accuracy) for our baseline models;
(2) Comparison between baseline models and models that include course evaluation and other instructor data;
(3) Fairness: Comparison between prediction of each academic quartile versus prediction of all students

6.1 Baseline Performance

Tables 1 and 2 show the baseline ability of each machine learning model to predict student success without any course evaluation data. These models were trained and tested on only basic course features, such as the term taken and number of students in the class, and student demographics.

	Passing			Potential		
Classifier	**F1**	**AUC**	**Acc**	**F1**	**AUC**	**Acc**
Gradient Boosting	0.666 ±0.053	0.869 ±0.034	0.814 ±0.029	0.790 ±0.014	0.770 ±0.012	0.721 ±0.014
Random Forest	0.640 ±0.046	0.865 ±0.030	0.808 ±0.019	0.800 ±0.012	0.776 ±0.016	0.730 ±0.015
AdaBoost	0.641 ±0.051	0.849 ±0.035	0.803 ±0.021	0.777 ±0.024	0.746 ±0.025	0.710 ±0.022
Decision Tree	0.637 ±0.050	0.825 ±0.034	0.798 ±0.020	0.787 ±0.021	0.737 ±0.024	0.710 ±0.024
Neural	0.623 ±0.064	0.853 ±0.032	0.795 ±0.032	0.779 ±0.007	0.764 ±0.027	0.712 ±0.010
SVC	0.598 ±0.049	0.835 ±0.037	0.794 ±0.021	0.782 ±0.009	0.743 ±0.032	0.697 ±0.014
Naive Bayes	0.488 ±0.217	0.783 ±0.035	0.719 ±0.024	0.029 ±0.026	0.683 ±0.034	0.383 ±0.009

Table 1: Predicting CS211 success—passing the class or achieving one's "potential" grade—from only student demographics and basic course features.

	Graduating		
Classifier	**F1**	**AUC**	**Acc**
Gradient Boosting	0.533 ±0.037	0.855 ±0.007	0.824 ±0.011
Random Forest	0.447 ±0.041	0.837 ±0.009	0.825 ±0.011
AdaBoost	0.563 ±0.047	0.839 ±0.028	0.823 ±0.023
Decision Tree	0.473 ±0.018	0.662 ±0.012	0.759 ±0.015
Neural	0.486 ±0.036	0.762 ±0.018	0.735 ±0.043
SVC	0.0 ±0.0	0.770 ±0.030	0.776 ±0.000
Naive Bayes	0.460 ±0.009	0.785 ±0.029	0.515 ±0.018

Table 2: Predicting a CS211 success measure—graduating with a CS degree—from only student demographics and basic course features.

6.2 Effect of Including Evaluation Data

Tables 3, 4, and 5 assess the difference in performance between the baseline models and those that incorporated evaluation and course data. The smallest p-values are in bold.

		Percentiles		Flags	
		t Statistic	p-value	t Statistic	p-value
All	1 Overall Eval	-0.0319	0.9754	0.6862	0.5120
	2 Overall Evals	0.6796	0.5176	0.5370	0.6059
	2 Full Evals	0.1160	0.9105	0.1773	0.8637
	Discrete IDs	0.3738	0.7191	0.3738	0.7191
Q1	1 Overall Eval	0.1700	0.8696	-0.1004	0.9227
	2 Overall Evals	-0.3412	0.7425	0.4117	0.6930
	2 Full Evals	-0.0476	0.9632	0.0834	0.9357
	Discrete IDs	-0.0594	0.9540	-0.0594	0.9541
Q2	1 Overall Eval	-0.2578	0.8034	-0.2765	0.7894
	2 Overall Evals	0.01077	0.9917	-0.3237	0.7549
	2 Full Evals	-0.1091	0.9161	-0.2489	0.8097
	Discrete IDs	0.0749	0.9421	0.0749	0.9421
Q3	1 Overall Eval	-0.3398	0.7430	-0.1586	0.8784
	2 Overall Evals	0.0113	0.9913	-0.0904	0.9302
	2 Full Evals	0.0334	0.9742	-0.0587	0.9551
	Discrete IDs	0.4269	0.6840	0.4269	0.6840
Q4	1 Overall Eval	-0.6431	0.5382	**-1.1410**	**0.2892**
	2 Overall Evals	0.6485	0.5379	0.5716	0.5864
	2 Full Evals	0.1852	0.8585	0.5546	0.5943
	Discrete IDs	-0.3446	0.7404	-0.3446	0.7404

Table 3: Experimental models' performance in predicting whether students passed CS211, versus baseline models.

		Percentiles		Flags	
		t Statistic	p-value	t Statistic	p-value
All	1 Overall Eval	-0.1432	0.8898	0.5000	0.6352
	2 Overall Evals	1.1452	0.2863	-0.5726	0.5831
	2 Full Evals	0.4472	0.6675	-0.6202	0.5549
	Discrete IDs	**1.5110**	**0.1695**	**1.5110**	**0.1695**
Q1	1 Overall Eval	-0.0160	0.9877	-0.1531	0.8822
	2 Overall Evals	0.2992	0.7728	0.0360	0.9722
	2 Full Evals	-0.2371	0.8186	0.0559	0.9569
	Discrete IDs	0.3335	0.7475	0.3335	0.7475
Q2	1 Overall Eval	0.2996	0.7724	-0.0688	0.9469
	2 Overall Evals	0.7417	0.4804	-0.1990	0.8473
	2 Full Evals	0.1707	0.8688	0.2541	0.8061
	Discrete IDs	1.1216	0.2947	1.1216	0.2947
Q3	1 Overall Eval	0.1039	0.9199	0.1922	0.8526
	2 Overall Evals	0.5626	0.5894	0.1778	0.8637
	2 Full Evals	0.7600	0.4716	-0.2082	0.8403
	Discrete IDs	**1.1218**	**0.2946**	**.1218**	**0.2946**
Q4	1 Overall Eval	-0.3295	0.7502	-0.4261	0.6815
	2 Overall Evals	0.2247	0.8279	-0.1384	0.8936
	2 Full Evals	0.3297	0.7512	-0.5162	0.6203
	Discrete IDs	0.2808	0.7862	0.2808	0.7862

Table 4: Experimental models' performance in predicting whether students achieved their "potential" grade in CS211, versus baseline models.

The **Percentiles** column indicates evaluation scores were converted to percentiles; the **Flags** column indicates binary flags of the top 10% of scores were used.[4]

	Experiment	t Statistic	p-value
All	1 Overall Eval	-0.6515	0.5334
	2 Overall Evals	-0.7630	0.4781
	2 Full Evals	0.2207	0.8318
	Discrete IDs	-0.5975	0.5669
Q1	1 Overall Eval	0.2620	0.8008
	2 Overall Evals	0.2671	0.7964
	2 Full Evals	0.4025	0.6982
	Discrete IDs	0.3470	0.7385
Q2	1 Overall Eval	-0.5466	0.5996
	2 Overall Evals	-0.7459	0.4787
	2 Full Evals	-0.3600	0.7284
	Discrete IDs	-0.1730	0.8670
Q3	1 Overall Eval	**-0.7931**	**0.4557**
	2 Overall Evals	0.0139	0.9893
	2 Full Evals	-0.1678	0.8714
	Discrete IDs	0.2165	0.8352
Q4	1 Overall Eval	-0.2212	0.8311
	2 Overall Evals	0.4312	0.6778
	2 Full Evals	0.5631	0.5889
	Discrete IDs	-0.5019	0.6293

Table 5: Experimental models' performance in predicting retention in the CS major, versus baseline models.

In all of these t-tests, our null hypothesis was that evaluations and specific courses taken by a student do not improve student success predictions. If this were true, results from the baseline set of data would be the same as results that included course information because the course information would add no predictive power. None of our experiments proved to have a significant improvement over our baseline, so we fail to reject our null hypothesis and do not find that evaluations improve predictions of student success.

6.3 Fairness Across Student Quartiles

Tables 6, 7, and 8 show fairness t-tests. These are tests of whether the performance of each experimental model is better or worse at predicting results for a specific quartile, versus predicting results for all students. They capture the statistical significance of discrepancies in performance when run on different groups of students.

The null hypothesis in these tests is that there is no difference between the F1 scores for all students and those of each quartile. In other words: the null hypothesis is that the predictions are fair. The lowest p-scores we found are in bold or, if they are statistically significant, are highlighted.

Table 6 shows the models' fairness in predicting whether students passed CS211.

Table 7 shows fairness in predicting whether students achieved their potential grades. This table differs much from Table 6 in that many of the p-values listed here are significant at the 0.05 level. All of the significant results are clustered within the first and second quartiles, which are the bottom two quartiles in our groupings.

Table 8 shows models' fairness in predicting whether students graduate with a CS major. While the significant t statistics in Table 7 were positive—indicating that the models perform best on the first and second quartiles—we see that performance for the lower two quartiles is negative. Additionally, t statistics are significantly

[4] Note that for models using Discrete IDs, we do not use numerical evaluation data, so there is no distinction between the two categories' results.

		Percentiles		Flags	
		t Statistic	p-value	t Statistic	p-value
Q1	**Baseline**	1.1476	0.2884	1.1476	0.2884
	1 Overall Eval	0.9143	0.3899	0.6463	0.5443
	2 Overall Evals	1.3646	0.2097	1.2095	0.2610
	2 Full Evals	1.1995	0.2669	**1.4597**	**0.1896**
	Discrete IDs	1.0130	0.3471	1.0130	0.3471
Q2	**Baseline**	-0.2019	0.8475	-0.2019	0.8475
	1 Overall Eval	-0.4798	0.6519	-0.2687	0.7957
	2 Overall Evals	-0.3005	0.7746	-0.3801	0.7179
	2 Full Evals	-0.1890	0.8568	-0.4047	0.6965
	Discrete IDs	0.3129	0.7640	0.3129	0.7640
Q3	**Baseline**	0.4792	0.6448	0.4792	0.6448
	1 Overall Eval	0.3567	0.7316	0.3720	0.7197
	2 Overall Evals	0.2940	0.7772	-0.0392	0.9700
	2 Full Evals	0.5200	0.6172	0.2183	0.8327
	Discrete IDs	0.5267	0.6133	0.5267	0.6133
Q4	**Baseline**	-0.6228	0.5560	-0.6228	0.5560
	1 Overall Eval	-0.3376	0.7453	-0.6068	0.5612
	2 Overall Evals	**-1.3819**	**0.2078**	-0.3570	0.730423
	2 Full Evals	-0.6432	0.5467	-0.5803	0.5793
	Discrete IDs	-0.7653	0.4773	-0.7653	0.4773

Table 6: Fairness in experimental models' predictions of whether students passed CS211.

		Percentiles		Flags	
		t Statistic	p-value	t Statistic	p-value
Q1	**Baseline**	2.5055	0.0557	2.5055	0.0557
	1 Overall Eval	1.9566	0.1005	1.7817	0.1338
	2 Overall Evals	1.9229	0.1069	2.9422	0.0278
	2 Full Evals	2.0311	0.0960	2.540979	0.0462
	Discrete IDs	2.0484	0.0858	2.0484	0.0858
Q2	**Baseline**	3.1270	0.0267	3.1270	0.0267
	1 Overall Eval	2.6275	0.0445	2.8256	0.0371
	2 Overall Evals	3.2574	0.0162	4.4184	0.0045
	2 Full Evals	4.5984	0.0019	3.2834	0.0196
	Discrete IDs	3.5205	0.0134	3.5205	0.0135
Q3	**Baseline**	0.0994	0.9246	0.0994	0.9246
	1 Overall Eval	0.4097	0.6943	0.3179	0.7602
	2 Overall Evals	0.3143	0.7647	-0.1453	0.8895
	2 Full Evals	0.2490	0.8111	0.6658	0.5346
	Discrete IDs	0.3603	0.7305	0.3603	0.7305
Q4	**Baseline**	-2.0964	0.1004	-2.0964	0.1004
	1 Overall Eval	-2.4168	0.0650	-2.4568	0.0627
	2 Overall Evals	-2.5259	0.0556	-2.3934	0.0695
	2 Full Evals	-2.1430	0.0907	-2.3262	0.0736
	Discrete IDs	-2.5112	0.0585	-2.5112	0.0585

Table 7: Fairness in experimental models' predictions of whether students achieved their "potential" grade in CS211.

better for students in the top quartile. This suggests significant fairness disparities in these prediction models: the F1 scores of prediction across the entire student body overlap strongly with the F1 scores across the third quartile, but vary widely from those of both stronger and poorer overall performers. This is in spite of the quartiles being represented in the overall dataset in equal numbers of data points.

	type	t Statistic	p-value
Q1	**Baseline**	-5.7063	0.0019
	1 Overall Eval	-3.4333	0.0195
	2 Overall Evals	-4.4204	0.0106
	2 Full Evals	-6.6634	0.0013
	Discrete IDs	-3.6290	0.0178
Q2	**Baseline**	-2.5044	0.0528
	1 Overall Eval	-2.7323	0.0380
	2 Overall Evals	-4.3013	0.0106
	2 Full Evals	-3.6916	0.0161
	Discrete IDs	-2.7675	0.0381
Q3	**Baseline**	-0.6357	0.5498
	1 Overall Eval	-1.4485	0.1859
	2 Overall Evals	-0.4967	0.6411
	2 Full Evals	-1.3347	0.2360
	Discrete IDs	-0.1085	0.9166
Q4	**Baseline**	2.4488	0.0421
	1 Overall Eval	3.1471	0.0144
	2 Overall Evals	4.4240	0.0072
	2 Full Evals	3.1356	0.0225
	Discrete IDs	2.4960	0.0409

Table 8: Fairness in experimental models' predictions of whether students graduated with a CS major.

7 DISCUSSION

7.1 Improvements Upon the Baseline

Overall, our results show that adding evaluations to predictions does not significantly improve predictions over the baseline of student demographics and basic course features. The addition of instructor gender, too, was not significant. Even when gender was included for all courses used in predictions, the results did not improve drastically. However, there were many semesters for which there were no female instructors available at all to teach a course, so there could not be any direct comparisons between students with male instructors and those with female instructors. Although there are slight improvements for some experiments beyond the baseline, notably those involving the discrete and continuous unique IDs, they do not reach a significance level of 0.05. This suggests that the impact of student evaluation data and instructor gender on student performance is not immediately visible.

Generally, our experiments performed better when predicting whether students would achieve their "potential" grades than whether they would receive credit. When comparing p-values between passing and potential for all experiments, as in Tables 4, 5, and 6, predicting student potential seems to improve upon predicting passing even when compared to their respective baselines. We attribute this to the fluid nature of a student's forecast grade: if a students forecast grade is a C, then there are 3 possible grades that this student could get and still fulfill at or above his or her potential. Similarly, for students whose predicted grades are A's, there is only one possible grade, an A, with which they can achieve at or above their potential. This imbalance on both sides of the forecast grades means the models can make an easier prediction of achieving below a potential grade because there are generally more options on the lower end of the grade scale than on the higher end.

In addition, because the evaluations we have access to are only averages for all students in a course and do not reflect each student's personal evaluation of the professor, each student in a section of a course would have the same evaluations. This large amount of overlap between students, who then experienced different outcomes with their success, seemed to negatively impact the models in experiments where full sets of evaluations were used. This issue was slightly assuaged with the use of unique IDs, but not at a significant level for most quartiles, see Tables 5, 6, and 7. For this reason, evaluation sets where evaluations are unique to each student would provide and interesting contrast to this work—individualized evaluations might provide high quality features for prediction.

7.2 Fairness

Our fairness results reach significance levels especially often in the third and fourth quartiles—see Tables 8, 9, and 10—which are the lower academic quartiles. These quartile models underperform against the models for all students, frequently significantly. This is a cause for concern–the students for which our models predict well on, quartiles 1 and 2, are the quartiles in which students often already perform well. There are a few reasons that might contribute to this underperformance on the lower quartile of students. One is that our quartiles are artificially created—although high school GPA and SAT scores are indicators of academic success in high school, they do not necessarily represent the same success in college. In addition, we split students into quartile depending on the percentile of their averaged HS GPA and SAT scores, not on any visible clusters within the data. These artificial clusters might not represent true student groups.

8 CONCLUSION

Our data suggests there is a pressing need to understand how students of different academic calibers experience the same curricula, given the disparities in their ultimate outcomes. It also suggests that evaluations of courses, at least as they are structured in our data, do not offer significant insights as to how a student will perform or whether he or she will remain in computer science. Lastly, we find that the starting course number or code may have some predictive power, suggesting that different courses may significantly impact the outcomes of students. The question now becomes one of identifying how we measure the different features of these courses.

There are several possible expansions on our methodology. As previously mentioned, our data is imbalanced, and using techniques such as cost-aware training, oversampling, undersampling, or Synthetic Minority Oversampling Technique [4] might enable a more balanced weighting of results and greater accuracy in identifying points in the minority classes. Another proposed fairness-oriented metric is the Absolute Between-ROC Area metric, which measures the absolute area between two ROC curves. In doing so, it measures disparities in prediction across every possible decision threshold, as opposed to just one[12]. Lastly, we would like to grid-search for hyperparameters that optimize F1 score and ROC AUC, rather than accuracy.

In addition to course evaluations for computer science classes, we also scraped course evaluations for other classes. In the future, we hope to use this dataset to apply such retention analysis to all majors. Given that GMU has unique student body, with many transfer students and non-traditional graduates, we would like to also include these students in a future analysis to track differences in their progressions through their majors. This also begs the question of whether our results would be different at a school with more four-year students. Despite the fact that our data does not indicate that evaluations can improve predictions of student success, we are interested in the outcomes of research into this avenue at schools with differing evaluation styles to see if these results can be improved upon. In addition, the fairness concerns raised in this paper around differing performances for students with varying academic statuses are of concern. We would like to see the improvement of grade prediction techniques both for all students and for each quartile or minority demographic.

ACKNOWLEDGMENTS

This work was funded by the National Science Foundation as a part of GMU's Computer Science Research Experience for Undergraduates, and graciously supported by Dr. Karen Lee and GMU's OSCAR office. We would also like to thank Drs. Mark Snyder and Huzefa Rangwala for their guidance and knowledge. Finally, we thank GMU for the use of their facilities and this opportunity.

REFERENCES

[1] E. P. Bettinger and B. T. Long. 2005. Do Faculty Serve as Role Models? The Impact of Instructor Gender on Female Students. *The American Economic Review* 95, 2 (2005), 152–157. http://www.jstor.org/stable/4132808

[2] Maureen Biggers, Anne Brauer, and Tuba Yilmaz. 2008. Student Perceptions of Computer Science: A Retention Study Comparing Graduating Seniors with Cs Leavers. *SIGCSE Bull.* 40, 1 (mar 2008), 402–406. https://doi.org/10.1145/1352322.1352274

[3] John A. Centra. 2003. Will Teachers Receive Higher Student Evaluations by Giving Higher Grades and Less Course Work? *Research in Higher Education* 44, 5 (2003), 495–518. https://doi.org/10.1023/A:1025492407752

[4] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[5] Chantal Cherifi, Hocine Cherifi, MÃąrton Karsai, and Mirco Musolesi (Eds.). 2018. *Complex Networks & Their Applications VI: Proceedings of Complex Networks 2017 (The Sixth International Conference on Complex Networks and Their Applications).* Studies in Computational Intelligence, Vol. 689. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-72150-7

[6] Peter A. Cohen. 1981. Student Ratings of Instruction and Student Achievement: A Meta-Analysis of Multisection Validity Studies. *Review of Educational Research* 51, 3 (1981), 281–309. https://doi.org/10.2307/1170209

[7] Paulo Cortez and Alice Silva. 2008. Using data mining to predict secondary school student performance. *EUROSIS* (01 2008).

[8] Benjamin J. Drury, John Oliver Siy, and Sapna Cheryan. 2011. When Do Female Role Models Benefit Women? The Importance of Differentiating Recruitment From Retention in STEM. *Psychological Inquiry* 22, 4 (2011), 265–269. https://doi.org/10.1080/1047840X.2011.620935

[9] Office of Institutional Effectiveness and Planning. 2019. *Degrees Conferred By Degree And Demographic - Year 2017-18, All Terms.* http://irr2.gmu.edu/New/N_Degree/DegDegreeDetail.cfm

[10] Office of Institutional Effectiveness and Planning. 2019. *Full-Time Academic Faculty Demographic Profiles Two-Year Comparisons.* http://irr2.gmu.edu/New/N_Faculty/FullTimeFacComp.cfm

[11] Kenneth A. Feldman. 1996. Identifying exemplary teaching: Using data from course and teacher evaluations. *New Directions for Teaching and Learning* 1996, 65 (Mar 1996), 41–50. https://doi.org/10.1002/tl.37219966509

[12] Josh Gardner, Christopher Brooks, and Ryan Baker. 2019. Evaluating the Fairness of Predictive Student Models Through Slicing Analysis. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge (LAK19).* ACM, New York, NY, USA, 225–234. https://doi.org/10.1145/3303772.3303791

[13] Jeff Kastner Gregory Warren Bucks, Kathleen A. Ossman and F James Boerio. 2015. First-year Engineering Courses' Effect on Retention and Workplace Performance. In *2015 ASEE Annual Conference & Exposition.* ASEE Conferences, Seattle, Washington. https://peer.asee.org/24114.

[14] Sara Morsy and George Karypis. 2017. Cumulative knowledge-based regression models for next-term grade prediction. *Proceedings of the 17th SIAM International Conference on Data Mining, SDM 2017* (jan 2017), 552–560. http://www.scopus.com/inward/record.url?scp=85027876583&partnerID=8YFLogxK

[15] Laird Townsend. 1994. How Universities Successfully Retain and Graduate Black Students. *The Journal of Blacks in Higher Education* 4 (1994), 85–89. http://www.jstor.org/stable/2963380

[16] Brian L. Yoder. 2017. Engineering by the Numbers. *Engineering College Profiles & Statistics Book* (2017). http://www.asee.org/papers-and-publications/publications/college-profiles/15EngineeringbytheNumbersPart1.pdf

A PREDICTION FEATURES

Table 9 shows the features used for each of the predictions, categorized by the type of experiment being described.

Features	Meaning	Baseline	Overall SET	All SET	IDs
Race	Categorical variable, includes option for no race listed.	×	×	×	×
Sex	Categorical variable: male, female, and no gender listed.	×	×	×	×
High School GPA	Continuous variable.	×	×	×	×
SAT Total Score	Continuous variable, out of 1600. Empty cells are filled in with the median of the total SAT scores.	×	×	×	×
SAT Verbal Score	Continuous variable, out of 800. Empty cells are filled in with the median of the total SAT verbal scores.	×	×	×	×
SAT Math Score	Continuous variable, out of 800. Empty cells are filled in with the median of the total SAT math scores	×	×	×	×
Average Percentile	Continuous variable, between 0 and 1. Average of the HS GPA and SAT Total percentiles for each student.	×	×	×	×
Class Term Taken	Continuous variable, indicates the term in which the student took the course used for prediction and the course being predicted.	×	×	×	×
Term GPA	Continuous variable, the non-cumulative GPA for the term in which the student took the course used for prediction and the course being predicted.	×	×	×	×
Instructor Gender	Binary variable, split between male and female.		×	×	×
Grade Points	Continuous variable, the grade received in the course used for predicting the second course.	×	×	×	×
Overall Evaluations	Continuous or binary, depending on the treatment of the specific test—flagging or percentiles. These are defined as SET (as seen in Appendix 9.2) questions 15 and 16.		×	×	
All Evaluations	Continuous or binary, depending on the treatment of the specific test—flagging or percentiles. These are defined as SET (as seen in Appendix 9.2) questions 1 through 14.			×	
Course ID	Binary, represents the unique course taken by a student: ID is discipline, course number, section number, term taken, and binary digit indicating a summer term. Students in the same course and section will all have a 1.				×

Table 9: The features used for each experiment.

B GMU'S STUDENT EVALUATION OF TEACHING (SET)

Each of these sections were rated on a scale of 1 to 5, with a NA option available. Questions 15 and 16 are the "overall" evaluations used in certain experiments.

(1) Course requirements and expectations were clear.
(2) The course was well organized.
(3) The instructor helped me to better understand the course material.
(4) Feedback (written comments and suggestions on papers, solutions provided, class discussion, etc.) was helpful.
(5) The instructor showed respect for the students.
(6) The instructor was accessible either in person or electronically.
(7) The course grading policy was clear.
(8) Graded work reflected what was covered in the course.
(9) The assignments (projects, papers, presentations, etc.) helped me learn the material.
(10) The textbook and/or assigned readings helped me understand the material.

(11) Assignments and exams were returned in a reasonable amount of time.
(12) The instructor covered the important aspects of the course as outlined in the syllabus.
(13) The instructor made the class intellectually stimulating.
(14) The instructor encouraged the students to be actively involved in the material through discussion, assignments, and other activities.
(15) My overall rating of the teaching.
(16) My overall rating of this course.

C EXTENDED GRADE PREDICTION RESULTS

			Percentiles			Top 10% Flags		
	type	**model**	**F1**	**AUC**	**Acc**	**F1**	**AUC**	**Acc**
Passing	1 overall	Gradient Boosting	0.665 ±0.046	0.872 ±0.034	0.809 ±0.029	0.689 ±0.053	0.875 ±0.032	0.826 ±0.032
	2 overall	Gradient Boosting	0.686 ±0.039	0.880 ±0.032	0.822 ±0.022	0.684 ±0.053	0.874 ±0.033	0.821 ±0.030
	2 full	Gradient Boosting	0.670 ±0.056	0.876 ±0.034	0.812 ±0.027	0.672 ±0.054	0.875 ±0.032	0.815 ±0.029
Potential	1 overall	Random Forest	0.799 ±0.010	0.770 ±0.018	0.728 ±0.013	0.803 ±0.006	0.773 ±0.017	0.734 ±0.008
	2 overall	Random Forest	0.808 ±0.010	0.787 ±0.018	0.741 ±0.013	0.796 ±0.010	0.773 ±0.017	0.725 ±0.014
	2 full	Random Forest	0.804 ±0.016	0.788 ±0.030	0.739 ±0.019	0.796 ±0.008	0.771 ±0.013	0.722 ±0.010

Table 10: Highest performing models from each of the experiments, evaluation treatments, and grade prediction styles. The best performers in each grade prediction style block are highlighted.

The experiment that improved upon the baseline power of prediction most utilizes unique course IDs to represent individual courses taken. The results of this type of experiment are displayed in Table 2. Table 2 contains the results of the ID experiments predicting student grades. The top performing models in Table 2 outperform the baseline predictive powers in F1, AUC, and accuracy measures, and the significance of these experiments is explored in Table 4.

	Passing			Potential		
Classifier	**F1**	**AUC**	**Acc**	**F1**	**AUC**	**Acc**
Gradient Boosting	0.677 ±0.039	0.876 ±0.029	0.821 ±0.015	0.811 ±0.013	0.801 ±0.015	0.748 ±0.017
AdaBoost	0.666 ±0.039	0.858 ±0.024	0.810 ±0.018	0.794 ±0.015	0.788 ±0.023	0.736 ±0.013
Neural Net	0.664 ±0.044	0.850 ±0.031	0.812 ±0.016	0.763 ±0.025	0.761 ±0.031	0.697 ±0.033
Random Forest	0.639 ±0.059	0.878 ±0.022	0.814 ±0.023	0.811 ±0.011	0.804 ±0.014	0.745 ±0.014
SVC	0.633 ±0.060	0.849 ±0.029	0.815 ±0.021	0.772 ±0.016	0.749 ±0.022	0.697 ±0.017
Decision Tree	0.605 ±0.068	0.802 ±0.047	0.800 ±0.021	0.789 ±0.019	0.739 ±0.015	0.719 ±0.025
Naive Bayes	0.485 ±0.005	0.550 ±0.010	0.377 ±0.014	0.284 ±0.051	0.673 ±0.032	0.466 ±0.021

Table 11: predicting passing 211 from unique IDs for both courses and using either continuous or discrete