Exploring techniques to improve machine learning's identification of at-risk students in physics classes

John Pace, John Hansen, and John Stewart[*]

Department of Physics and Astronomy, West Virginia University, Morgantown, West Virginia 26506, USA

(Received 6 January 2024; accepted 16 May 2024; published 31 May 2024)

Machine learning models were constructed to predict student performance in an introductory mechanics class at a large land-grant university in the United States using data from 2061 students. Students were classified as either being at risk of failing the course (earning a D or F) or not at risk (earning an A, B, or C). The models focused on variables available in the first few weeks of the class which could potentially allow for early interventions to help at-risk students. Multiple types of variables were used in the model: in-class variables (average homework and clicker quiz scores), institutional variables [college grade point average (GPA)], and noncognitive variables (self-efficacy). The substantial imbalance between the pass and fail rates of the course, with only about 10% of students failing, required modification to the machine learning algorithms. Decision threshold tuning and upsampling were successful in improving performance for at-risk students. Logistic regression combined with a decision threshold tuned to maximize balanced accuracy yielded the strongest classifier, with a DF accuracy of 83% and an ABC accuracy of 81%. Measures of variable importance involving changes in balanced accuracy identified homework grades, clicker grades, college GPA, and the fraction of college classes successfully completed as the most important variables in predicting success in introductory physics. Noncognitive variables added little predictive power to the models. Classification models with performance near the best-performing models using the full set of variables could be constructed with very few variables (homework average, clicker scores, and college GPA) using straightforward to implement algorithms, suggesting the application of these technologies may be fairly easy to include in many physics classes.

DOI: 10.1103/PhysRevPhysEducRes.20.010149

I. INTRODUCTION

Quantitatively understanding student outcomes in physics classes has long been an important research strand in physics education research (PER). Quantitative studies have explored how course grades, final examination scores, and conceptual inventory scores change with student characteristics and with different methods of instruction. In general, these studies have applied traditional statistical methods including linear and logistic regression. Recently, a wide variety of new computational data analysis techniques broadly classified as machine learning has been created to analyze the large datasets collected by public and private entities in the Internet age. These methods are rarely based on underlying statistical assumptions and often do not allow statistical conclusions. These methods have begun to be applied to examine student outcomes in physics classes [1,2] and physics student retention [3]. The current work explores this wealth of new methods and develops recommendations for the use of these for accurate prediction of all students with minimum additional effort for working instructors. Predicting student outcomes in physics classes could be a key step in improving methods of instruction as well as decreasing the attrition rate of students from science, technology, engineering, and mathematics (STEM) degrees. The President's Council of Advisors on Science and Technology issued a report [4] in 2012 that called for an increase of graduates in STEM majors so as to avoid a shortfall of one million STEM job candidates over the next decade. An accurate prediction of a student's outcome in a physics class allows instructors to target students who are at risk of failing the class with interventions that could improve their chance of passing the class. Physics classes form key barriers to degree progression for STEM students and, therefore, improving physics retention should improve STEM retention in general.

A. Research questions

This study seeks to further explore the application of machine learning algorithms to predict whether a student will earn a D or F in a physics class, to understand how the

[*] jcstewart1@mail.wvu.edu

accuracy of these predictions are affected by different algorithms and additional variables including noncognitive variables, and to understand what these algorithms can contribute to general research landscape of PER. In particular, we explore the following research questions:

RQ1: How can machine learning outcomes be optimized to most effectively predict student outcomes early in physics classes?

RQ2: How does the performance of the algorithms change with the addition of new types of variables such as noncognitive or institutional variables? What factors are most important in the prediction of student success in physics classes?

RQ3: How does the performance of the optimized model compare with that of a model using a limited set of variables easily accessible to physics instructors?

B. Prior machine learning studies in physics

The application of machine learning techniques to student performance prediction in PER has become more common in recent years. A 2018 study by Aiken *et al.* used random forest classifiers to examine student persistence in physics and which factors affect student transitions into engineering while also comparing the results to those of a more traditional contingency table analysis [3]. Both methods indicated that the two features most useful in predicting whether a student would complete a physics degree or transfer to an engineering degree were whether a student had taken a modern physics course or had enrolled in engineering courses while listed as a physics major. The random forest classifier achieved a ROC-AUC score of 0.93, indicating a stronger predictive performance than random chance guessing.

The current work builds on two prior studies focusing on the early identification of students at risk of failing or withdrawing from an introductory physics course [1,2]. In the first study, Zabriskie *et al.* [1] used random forest and logistic regression classifiers to predict students at risk of earning a grade lower than B in both introductory mechanics and electromagnetism courses using a combination of institutional and in-class variables. Grades of A or B rather than A, B, and C were predicted to approximately balance the two outcomes. The best-performing models achieved a classification accuracy of 73% for physics 1 and 81% for physics 2 using only data available in the first week of class; performance improved in subsequent weeks as additional student performance data were added. The overall classification accuracy is the fraction of predictions that are correct. Multiple metrics characterizing classification algorithms are discussed in Sec. II E; classification accuracy is defined in Eq. (1) in this section. Variable importance measures from the random forest classifiers indicated that college GPA was the most important variable for the model's classifications until the addition of the student's first exam score in week 5 of the course. Students' average homework scores and percentage of successfully completed credit hours relative to total attempted credit hours were also consistently found to be important to model performance.

In the second study, Yang *et al.* [2] focused only on the mechanics course and explored a slightly different classification goal, attempting to classify students likely to either receive a grade of D or F or who would withdraw (W) from the course. This resulted in an unbalanced dataset, with only 10%–20% of the students across various datasets failing or withdrawing. The sample balance is the ratio of the students earning D, F, or W to those earning A, B, or C. Random forest classifiers with decision thresholds tuned to increase performance on the minority class were constructed, and classification accuracies were examined for both the ABC and DFW categories. The best-performing model achieved a total accuracy of 91% with a DFW accuracy of 53% and an ABC accuracy of 95% using both institutional variables and in-class variables available in the first week of class. Again, model performance improved with time as additional student performance data were gathered. Institutional variables were found to be substantially more important than in-class variables early in the semester. Students' college GPA, average homework grades, and percentage of completed credit hours were found to be the most important variables for improving DFW accuracy, similar to the first study. An analysis of model performance on demographic subgroups on two datasets from different institutions indicated that models trained on predominantly majority students performed equally well on women, PEER (persons excluded due to ethnicity or race), and first-generation college students (FGCS), with all performance differences being within one standard deviation of the model's overall performance.

This work expands on these prior studies by exploring new techniques to account for heavy sample imbalance and improve the identification of DF students. An alternative optimization metric robust against sample imbalance, balanced accuracy (Sec. II E), was introduced. This new metric dramatically improved the accuracy of the identification of unsuccessful students. This study investigated a substantially larger set of variables than prior studies including noncognitive variables and variables related to students' high school physics preparation. Variable importance analyses were extended to improve model interpretability, inform future model construction, and identify minimal subsets of variables that result in near-optimal performance.

C. Educational data mining and learning analytics

Educational data mining (EDM) is a broad field involving the use of statistical, machine learning, and traditional data mining techniques to analyze and interpret educational data. The field has experienced rapid growth recently, in part due to increasingly prevalent large-scale data collection through platforms such as learning management systems and intelligent tutoring systems. In addition to more standard techniques, EDM studies have used less traditionally applied methods such as psychometric modeling [5]. A 2014 review of 240 studies by Peña-Ayala found that 88% used some form of statistical or machine learning approach to draw conclusions from their data [6]. A 2019 review by Aldowah *et al.* of 402 EDM studies found that 63% focused on computer-supported predictive analytics, with a main focus on evaluating and monitoring student learning. The same review also found that 26% of studies used classification, 21% clustering, and 10% regression [7].

A variety of statistical and machine learning methods, including logistic regression, decision trees, random forests, neural networks, naive Bayes, support vector machines, and K-nearest neighbor algorithms have been used within EDM [8]. More information on these and other machine learning techniques can be found in several machine learning texts [9,10] and in the Supplemental Material [11].

Learning analytics (LA) focuses on gathering and analyzing data related to learners and the learning environment with the goal of understanding and improving the learning process and environment. While EDM focuses on the technological challenge of analyzing large datasets and developing new methods and models, LA focuses more on the application of known predictive models to decision making [5]. LA is a broad field encompassing many topics including the prediction of student outcomes. In addition to predicting student performance in courses, outcomes such as the retention of PEER students in STEM [12] and students' participation in STEM careers [13] have been explored. Such predictions can be used in early warning systems to aid student performance and retention and to understand what motivates performance gaps between different groups of students [14].

LA also seeks to understand how students navigate their courses, often by using data gathered either within the classroom or by a course's learning management system. Rose *et al.* explored the use of explanatory learner models and discussed associated best practices, with goals of predicting student outcomes and offering actionable insights as to how students interact with the material and structure of a course [15]. Spikol *et al.* developed a multimodal system utilizing computer vision, audio, and Arduino IDE data to track student progress through a project and identify which features of students' group work predicted project outcomes [16]. Process mining models of how students use and navigate course learning management systems using activity logs have been constructed and have been used to compare how passing and failing students interact with the course [17].

D. Grade prediction and persistence

Grade prediction is a common goal within EDM and LA; machine learning techniques are commonly used for this purpose. A 2020 review of 64 articles focusing on predicting student behavior using machine learning by Rastrollo-Guerrero *et al.* found that 70% focused on the prediction of student performance. The two most common methods employed in these works were support vector machines and neural networks; decision trees, random forests, naive Bayes, logistic regression, and K-nearest neighbors were also frequently utilized [18]. Grade prediction has also been explored for distance learning courses [19] and a sophomore-level engineering course [20].

Such predictive tasks are often performed using heavily unbalanced datasets, because courses generally have much higher pass rates than fail rates. Methods to improve performance on such unbalanced datasets have been explored in various studies, such as upsampling, decision threshold tuning, and feature selection [21]. Some of these methods are explored in this work.

II. METHODS

A. Sample

The sample was collected in the introductory calculus-based mechanics class at a large eastern land-grant university in the United States. The overall undergraduate student population in 2019 was 20 500 students with demographic composition 82% White, 4% African American, 4% Hispanic, 4% international, 4% two or more races, with other groups representing 2% or less. The 25th percentile to the 75th percentile range of ACT composite scores was 21 to 27 [22]. For the class studied, the demographic composition was 75% White, 14% international, 4% two or more races, 2% African American, 2% Asian, 2% Hispanic, with other groups representing 2% or less. The sample was collected from the Spring 2017 to the Fall 2019 semester. The class was presented with three 50-min lectures per week and one 3-h required laboratory session. During this time, the class was managed by a single lead instructor who oversaw general class policy, laboratory activities, and homework. This instructor was knowledgeable about PER and instructed many of the lecture sections; other faculty and staff instructed the other sections in collaboration with the lead instructor. The class was taught in two to three large lecture section each semester enrolling between 50 and 160 students each. All sections used Mazur's peer instruction pedagogy using clickers in the lecture [23]. The laboratory sessions featured a combination of hands-on inquiry based activities, whiteboarding exercises, traditional laboratories, and group problem-solving activities. There were many lab sections each semester each enrolling a maximum of 24 students.

B. Variables

The variables used in this work are introduced in Table I. The variables used are a combination of commonly

TABLE I. List of variables. Type indicates whether the variable is continuous (C) or dichotomous (D). Variables in a Panel must be used as a group. Variables marked BL are the base level of a group of variables in a panel. Datasets are labeled D_0 to D_3; see Sec. II C for a description of each.

Abbreviation	Type	Panel	BL	D_0	D_1	D_2	D_3	Description
TstAve	C			×	×	×	×	Test average.
HasGPA	D			×	×			Has college GPA.
Hwk2	C			×	×	×	×	Homework average by Week 2.
Clicker2	C			×	×	×	×	Total clicker percentage score Week 2 (percentage of lectures attended).
HasSurveys	D			×	×	×		Both surveys completed.
PreTaken	D			×				FMCE pretest completed.
Pretest	C				×	×	×	FMCE pretest percentage.
Complete	C					×	×	Percentage of classes completed before class.
CGPA	C					×	×	College grade point average before class.
STEMCls	C			×	×	×	×	STEM classes completed before class.
Credit	C			×	×	×	×	Credit hours completed before class.
Enroll	C			×	×	×	×	Current hours enrolled in semester of physics class.
ACTM	C					×	×	ACT or SAT mathematics percentile score.
ACTV	C					×	×	ACT English or SAT verbal percentile score.
HSGPA	C					×	×	High school grade point average.
APCNoMP	C			×	×	×	×	Number of non-math or non-physics AP classes with college credit.
APPhys	D			×	×	×	×	Credit for AP Physics.
APCalc	D			×	×	×	×	Credit for AP Calculus.
HSP.NTake	D	×	×					High school physics not taken.
HSP.NAP.NA	D	×					×	High school physics class not AP—grade B, C, D.
HSP.NAP.A	D	×					×	High school physics class not AP—grade A.
HSP.APNP.NA	D	×					×	High school physics AP (test not passed)—grade B, C, D.
HSP.APNP.A	D	×					×	High school physics AP (test not passed)—grade A.
HSP.APP.NA	D	×					×	High school AP physics (test passed)—grade B, C, D.
HSP.APP.A	D	×					×	High school AP physics (test passed)—grade A.
MathReady	D			×	×	×	×	Was the student's first college math class Calculus 1 or higher?
HSMathA	D						×	Was the grade in most advanced high school math class an A?
HSMNotCal	D	×	×					Was most advanced high school math class below calculus?
HSMNotAP	D	×					×	Was most advanced high school math class calculus?
HSMAPNPass	D	×					×	Was most advanced high school math class AP calculus (test not passed)?
HSMAPPass	D	×					×	Was most advanced high school math class AP calculus (test passed)?
TRCNoMP	C			×	×	×	×	How many non-math and non-physics transfer classes?
TRPhys	D			×	×	×	×	Does the student have transfer credit for physics?
TRMath	D			×	×	×	×	Does the student have transfer credit for math?
Belong	C						×	Sense of belonging in physics class.
SelfEff	C						×	Self-efficacy towards physics class.
GrdExA	D	×					×	Does the student expect to earn an A in physics?
GrdExB	D	×					×	Does the student expect to earn a B in physics?
GrdExCDFW	D	×	×					Does the student expect to earn a C, D, F, or W in physics?
Agr	C						×	BFI personality facet—Agreeableness.
Cns	C						×	BFI personality facet—Conscientiousness.
Nrt	C						×	BFI personality facet—Neuroticism.
Ext	C						×	BFI personality facet—Extraversion.
Opn	C						×	BFI personality facet—Openness.
Repeat	D			×	×	×	×	Is the student repeating the class?
Gender	D			×	×	×	×	Does the student identify as female?
FirstGen	D			×	×	×	×	Is the student a first-generation college student?
PEER	D			×	×	×	×	Does the student identify as PEER?

available in-class variables such as Force and Motion Conceptual Evaluation (FMCE) [24] pretest score and homework average, variables that can be obtained from the institution such as college grade point average (CGPA) and demographic characteristics, and variables collected for this study by the application of two survey instruments during the class. The variables are organized into datasets by their availability; the variables available in each dataset are shown in Table I. Variables are labeled as continuous (C) or dichotomous (D). Some dichotomous variables such as the type of high school physics taken are the result of dummy coding a multilevel categorical variable; these are marked as being in a "Panel." The set of variables in a panel are not independent; one variable can be exactly calculated from all other variables. To use the panel of variables in a regression, one variable must be eliminated to remove this dependency; that variable becomes the base level of the panel. The regression coefficients of other variables in the panel measure changes with respect to the base level.

1. In-class variables

Some form of in-class variables should be available to all physics instructors. For the class studied, clickers are used to implement the Peer Instruction pedagogy [23] and are graded for participation. The clicker average score is a measure of attendance and is available as a running average throughout the semester. This work used the clicker average (Clicker2) at the end of the second week of class. Homework is collected weekly; the homework average score at the end of the second week was also used (Hwk2). These variables are measured in the second week of class because this is after the add deadline and a homework has been reliably collected by this time. The variables Hwk2 and Clicker2 are available for all students. The class administers the force and motion conceptual evaluation during the first week of class; the modified scoring rubric suggested by Thornton *et al.* was used in this work [25]. Not all students complete the FMCE; as such, pretest scores are only available for a subset of students; however, for all students, a dichotomous variable PreTaken measuring whether the pretest was taken is available.

2. Institutional variables

Colleges maintain a detailed set of information on all students. This study accessed a subset of this information, called institutional variables, through a request to the university. The variables requested include demographic variables (gender, FGCS status, PEER status) [26], high school preparation measures (high school GPA (HSGPA), ACT and SAT mathematics and verbal scores, and Advanced Placement and transfer credit), and college academic measures (college GPA and credits completed). All students were first-time freshmen; as such, transferred classes would have been taken as dual enrollment in high school. A student is considered to have PEER status if they report a race different than White or Asian or an ethnicity different than non-Hispanic/Latino.

3. Survey instruments

Additional information was collected from the students through two online surveys given early in the semester. Participation in the survey was incentivized by a small amount of course credit. The first survey collected detailed information about the student's high school science and mathematics classes. The second survey measured a set of noncognitive variables: self-efficacy, sense of belonging, and personality. Self-efficacy was measured using the self-efficacy for learning and performance subscale from the Motivated Strategies for Learning Questionnaire (MSLQ) [27]. Sense of belonging was measured with three items from Good *et al.* "Math Sense of Belonging" instrument [28]. Personality was measured with the Big Five Inventory (BFI) which measures the five-factor model of personality with factors: agreeableness, conscientiousness, extraversion, neuroticism, and openness [29–31]. These noncognitive constructs have been used in many studies examining college academic achievement [32].

C. Datasets

This work explores the prediction of student physics grades using a sequence of nested datasets. Table I shows the set of variables included in each dataset. For each dataset, students without a value for all variables have been removed. For all datasets, any continuous variables were standardized.

1. Dataset 0 (D_0)

Dataset 0 (D_0) contains variables generally available for all students who complete the course for a grade. Some in-class data such as clicker scores or homework grades were missing for some course sections; students without this information were removed. D_0 contains $N_0 = 2061$ complete records. Students withdrawing from the course did not have homework averages recorded because this information was stored in the learning management system and students were removed from this system when they withdrew. D_0 contains variables measuring demographics, AP and transfer credit, math readiness, whether the student was repeating the class, and three variables indicating the availability of additional data: HasGPA, HasSurveys, and PreTaken. D_0 also contains some college-level variables measuring credit earned and current credits enrolled. A student is considered "math ready" if they are prepared to enroll in calculus 1 or a more advanced mathematics class in their first semester.

2. Dataset 1 (D_1)

Dataset 1 (D_1) contains $N_1 = 1870$ complete records and includes students in D_0 with a FMCE pretest score. The variable PreTaken is no longer useful in this dataset.

3. Dataset 2 (D_2)

Dataset 2 (D_2) contains $N_2 = 1602$ complete records; this dataset contains college-level achievement variables including CGPA and the percentage of classes attempted that were completed for a grade. This dataset removes students taking the class in their fall freshman semester from D_1 because they do not have a meaningful college GPA yet. The variable HasGPA is no longer useful for this dataset. The dataset is also restricted to students with basic high-school-level variables such as high school GPA and standardized test scores (ACT or SAT).

4. Dataset 3 (D_3)

Dataset 3 (D_3) contains $N_3 = 1210$ complete records; this dataset removes students from D_2 who did not complete the two optional surveys. The variable HasSurveys is no longer useful for this dataset.

D. Descriptive statistics

The sequence of nested datasets D_0 to D_3 described above contains different numbers of students; as the number of records changes, the relative balance between the ABC and DF cases also changes. Sample balance can affect some classification statistics. The general characteristics of the students also change. In D_1, all students were in attendance the day the FMCE pretest was given; in D_2, all students had completed a semester of college; in D_3, all students completed two optional assignments (showing they were paying attention, communicating, and were willing to do an optional assignment to improve their grades). Table II presents descriptive statistics for the four datasets. For continuous variables, the mean ± standard deviation is reported. For dichotomous variables, the percentage of students in the high level of the variable (students possessing that feature) is reported.

E. Classification algorithms

A classification algorithm makes predictions on a dataset by assigning each case (student) in the dataset to one of two levels called the positive and negative levels. To build a classifier, a machine learning algorithm is selected, trained, and optimized. To do this, the original dataset is randomly

TABLE II. Descriptive statistics. Continuous variables are reported as mean ± standard deviation. For dichotomous variables, the fraction of students in the high level of the variable is reported.

	Dataset			
Variable	D_0	D_1	D_2	D_3
Test average	69.0 ± 16.1	70.0 ± 15.8	70.8 ± 15.1	72.1 ± 14.5
Physics grade DF, (DF = 1)	15.0%	12.4%	11.7%	7.8%
Has CGPA	95.4%	95.2%	100%	100%
Week 2 homework percentage	85.5 ± 19.6	86.4 ± 18.3	86.1 ± 18.2	89.2 ± 14.3
Week 2 clicker percentage	84.3 ± 29.9	86.4 ± 27.9	87.0 ± 27.4	90.1 ± 23.8
Has surveys	70%	75.4%	75.7%	100%
FMCE pretest completed	90.7%	100%	100%	100%
FMCE pretest percentage		23.7 ± 19.7	24.0 ± 19.2	23.5 ± 19.3
College course completion percentage			92.2 ± 12.8	93.6 ± 11.5
College GPA			3.2 ± 0.5	3.3 ± 0.5
College credit earned	28.3 ± 17.7	27.3 ± 17.3	27.6 ± 16.2	27.4 ± 16.0
ACT or SAT mathematics percentage			80.7 ± 14.5	81.1 ± 14.0
ACT or SAT verbal percentage			74.2 ± 18.0	75.2 ± 17.7
High school GPA			3.8 ± 0.5	3.9 ± 0.4
Entered college math in calculus	59.4%	61.6%	64.0%	64.4%
Sense of belonging in physics				4.1 ± 0.7
Self-efficacy toward physics				4.1 ± 0.7
Physics grade expectation A				42.3%
Physics grade expectation B				40.7%
Is repeating physics class?	9.4%	7.1%	7.0%	5.3%
Gender (Female = 1)	21.3%	21.9%	23%	27.4%
First-generation (FirstGen = 1)	18.6%	18.0%	16.9%	16.7%
PEER (PEER = 1)	8.9%	8.8%	8.8%	7.7%
N	2061	1870	1602	1210

TABLE III. Confusion matrix.

	Actual negative	Actual positive
Predicted negative	True negative (TN)	False negative (FN)
Predicted positive	False positive (FP)	True positive (TP)

split into two nonoverlapping subsets, the training and the test dataset, with stratification to preserve the sample balance (the fraction of DF students in the full dataset). The test dataset is not examined during the training and optimization process. The optimized algorithm is then used to predict each case in the test dataset. The accuracy of the prediction is summarized by the confusion matrix shown in Table III.

In this work, we attempt to identify students likely to be unsuccessful in an introductory mechanics course; students predicted to earn a D or F form our positive classification. This leads to a very unnatural terminology where the positive classification is the unfavorable outcome; as such, we will work with a confusion matrix specialized to our classification problem as shown in Table IV. A classifier fundamentally has two success rates: $\alpha_{\rm ABC}$ the rate at which a new case who will earn an A, B, or C is classified as earning an A, B, or C and $\alpha_{\rm DF}$ the rate at which a new case who will earn a D or F is classified as earning a D or F. These two rates can be calculated from the confusion matrix. Let N be the total size of the test dataset; $N = T_{\rm DF} + F_{\rm DF} + T_{\rm ABC} + F_{\rm ABC}$. The total number of ABC cases in the test dataset is $N_{\rm ABC} = T_{\rm ABC} + F_{\rm DF}$; the total number of DF cases is $N_{\rm DF} = T_{\rm DF} + F_{\rm ABC}$. As such $\alpha_{\rm ABC} = T_{\rm ABC}/N_{\rm ABC}$ and $\alpha_{\rm DF} = T_{\rm DF}/N_{\rm DF}$; in the machine learning literature, $\alpha_{\rm DF}$ is called the sensitivity and $\alpha_{\rm ABC}$ the specificity. The confusion matrix can be calculated from N, $\alpha_{\rm ABC}$, $\alpha_{\rm DF}$, and one additional parameter, the sample balance. The sample balance, γ, is the ratio of the size of the DF class to the ABC class, $\gamma = N_{\rm DF}/N_{\rm ABC}$. In this study, the samples are substantially unbalanced ($N_{\rm DF} \neq N_{\rm ABC}$) with $\gamma \approx 0.1$ for each dataset.

Beyond $\alpha_{\rm ABC}$ and $\alpha_{\rm DF}$, prior studies calculated a number of additional performance statistics: the overall accuracy [Eq. (1)] [1] and the positive predictive value (PPV) [Eq. (2)] [2]. The overall accuracy is the fraction of predictions of either the DF or ABC cases which are correct.

$$\text{Overall accuracy} = \frac{T_{\rm DF} + T_{\rm ABC}}{N} = \frac{\alpha_{\rm ABC} + \gamma \cdot \alpha_{\rm DF}}{1 + \gamma}. \tag{1}$$

TABLE IV. Confusion matrix for this study.

	Actual ABC	Actual DF
Predicted ABC	True ABC ($T_{\rm ABC}$)	False ABC ($F_{\rm ABC}$)
Predicted DF	False DF ($F_{\rm DF}$)	True DF ($T_{\rm DF}$)

PPV is the fraction of the DF predictions that are correct. Note that this is affected by both correct predictions for the DF case and incorrect predictions for the ABC case.

$$\text{PPV} = \frac{T_{\rm DF}}{T_{\rm DF} + F_{\rm DF}} = \frac{\gamma \cdot \alpha_{\rm DF}}{\gamma \cdot \alpha_{\rm DF} + (1 - \alpha_{\rm ABC})}. \tag{2}$$

Note that both the overall accuracy and the PPV depend on the sample balance and are, therefore, influenced by oversampling and undersampling methods. This relation to sample balance can obscure large differences in $\alpha_{\rm ABC}$ and $\alpha_{\rm DF}$ if overall accuracy or PPV is used as the metric optimized in the training process. In this work, we explore optimizing an alternate metric not dependent on sample balance, the balanced accuracy $\bar{\alpha}$. The balanced accuracy is the average of the two success rates.

$$\text{Balanced accuracy} = \bar{\alpha} = \frac{\alpha_{\rm ABC} + \alpha_{\rm DF}}{2}. \tag{3}$$

In general, as part of the optimization process for a machine learning model, the model is tuned to optimize some performance metric. Most machine learning algorithms calculate the probability of the DF outcome for each student. To convert this probability into a classification, a "decision threshold," η, is selected. Students with probability above η are assigned to the DF class; those with probability below η to the ABC class. The default value of η is usually $\eta = 0.5$. This default value was used in Zabriskie *et al.* [1], producing an acceptable overall accuracy. However, $\alpha_{\rm ABC} \gg \alpha_{\rm DF}$, indicating that the classifier was much better at identifying A, B, and C students than D and F students. In Yang *et al.* [2], the decision threshold was tuned until $\alpha_{\rm DF} = \text{PPV}$ so that the fraction of the D and F students who were identified was set equal to the fraction of the D and F predictions that were correct. While a fairly intuitive criterion balancing the rate of identification with the accuracy of identification, this tuning sets $\gamma = (1 - \alpha_{\rm ABC})/(1 - \alpha_{\rm DF})$. For very unbalanced samples, this also results in $\alpha_{\rm ABC} \gg \alpha_{\rm DF}$. In this work, models are tuned to maximize balanced accuracy, $\bar{\alpha}$; the maximum of $\bar{\alpha}$ generally occurs near $\alpha_{\rm ABC} = \alpha_{\rm DF}$, producing a classifier that is equally accurate for all students.

One way to classify students is simply to guess. A pure guessing classifier has $\alpha_{\rm ABC} = \alpha_{\rm DF} = 0.5$ yielding $\bar{\alpha} = 0.5$. One might also simply guess that all students were in the majority case (ABC) yielding $\alpha_{\rm ABC} = 1$ and $\alpha_{\rm DF} = 0$ for a balanced accuracy of $\bar{\alpha} = 0.5$. This all-majority guessing scheme would produce an overall accuracy [Eq. (1)] close to 0.9 due to the sample balance $\gamma \approx 0.1$, indicating its lack of utility as an evaluation metric for this case. The use of balanced accuracy effectively weights correct prediction of the minority class more heavily, inversely to the sample balance, causing the model to equally prioritize performance on both classes on the training dataset. When comparing the models constructed in this paper, one should consider their improvement over the pure guessing classifiers.

F. Logistic regression classifiers

Machine learning offers a wealth of classification algorithms; the algorithm most used in PER is logistic regression (LR). Additional algorithms are discussed in the Supplemental Material [11]. Logistic regression is similar to linear regression where a continuous dependent variable is predicted by a linear combination of independent variables; however, in logistic regression, the log odds of a dichotomous outcome is predicted using a linear combination of independent variables. The odds of an event happening is the ratio of the probability the event happens P divided by the probability that it does not happen $1 - P$; odds $= P/(1 - P)$. The probability of the event can be calculated from the log odds and the results converted into a classification prediction by selecting a decision threshold η where the DF outcome is predicted if $P > \eta$. Such a classifier can be implemented in the "R" programming language; a working example of a logistic regression classifier is shown in the Supplemental Material [11].

As in Yang *et al.* [2], decision threshold tuning (DTT) can be used to adjust the decision threshold to produce models with improved performance. Yang *et al.* [2] adjusted models by modifying the decision threshold until α_{DF} was equal to the PPV; this, however, still produced models where α_{DF} was much smaller than α_{ABC}. In the current work, model parameters are adjusted ("tuned") until $\bar{\alpha}$ is maximized which generally produces models where the ABC success rate, α_{ABC}, and the DF success rate, α_{DF}, are approximately equal. DTT is needed because of the unbalanced nature of the sample. If only 10% of the students receive a D or F, then the default classification probability for LR of 0.5 is much too large; it should be commensurate with the actual probability of receiving a D or F.

A potential issue in the use of machine learning models is the possibility of the model "overfitting" to the training data, that is, overly optimizing model performance on the training data at the expense of worse generalization to unseen data. The high multicollinearity of educational datasets may make some variables redundant; the machine learning algorithm can use these redundant variables to specialize the model to small groups of students in the training dataset. A commonly used method to reduce overfitting of LR models is regularization which penalizes large regression coefficients. For this work, L2 regularization was used which introduces an additional term to the model's loss function that penalizes it according to the L2 norm of the variable weights (regression coefficients) in the linear model. The strength of this regularization is controlled by a coefficient treated as a model hyperparameter. The base implementation of LR in "R" does not include the option of regularization, but the base implementation in the Python library scikit-learn does.

Other machine learning methods were explored in this study and their performance was described in the Supplemental Material [11], but LR was ultimately selected for a combination of its relatively fast training times, better performance, and higher familiarity within PER. For more details, see the discussion of RQ1 in Sec. IV A.

G. SMOTE upsampling

DTT adjusts the classification threshold probability to compensate for the unbalanced ABC and DF outcomes. Other methods exist to address this problem. While DTT successfully mitigates problems introduced by sample imbalance for LR classifiers, it does not work for all machine learning algorithms. Support vector machine classifiers, for example, do not generate any probabilities and, therefore, DTT cannot be used. One method, applied unsuccessfully in other PER works on grade prediction [1,2], is the synthetic minority oversampling technique (SMOTE) that manufactures new minority cases (DF students) by interpolating between small groups of existing cases. SMOTE creates synthetic minority class data points by drawing lines between a minority data point and its nearest k minority class neighbors in the vector space of features and by randomly picking a point from one of these lines as a new synthetic data point. By creating artificial DF cases, SMOTE can produce a balanced dataset.

H. Variable importance

One natural method to prevent overfitting, and the loss of predictive accuracy in the test dataset which comes with it, is to restrict the number of variables to those most important to making the prediction. The identification and use of only important variables is called feature selection; to perform feature selection, one must calculate a measure of "variable importance." Many feature selection algorithms exist; some simply examine whether the variable is significant in the logistic regression, while others set a threshold for the amount of additional variance that must be explained for retention in the model.

Machine learning using classification introduces a number of measures of variable importance. These measures can help identify which variables provide the model with the most discriminatory power between the classes and inform future model construction and data collection. The first measure utilized in this study, first-in variable importance, represents the change in balanced accuracy from a null model containing only the intercept to a model containing an intercept and the variable. The null model has $\bar{\alpha} = 0.5$ because it guesses the student's classification, therefore, $\alpha_{\mathrm{DF}} = 0.5$ and $\alpha_{\mathrm{ABC}} = 0.5$. For example, CGPA has a first-in importance of 0.25, which means a model containing only CGPA has $\bar{\alpha} = 0.75$. The second measure, sampled variable importance, represents the average change in balanced accuracy when the feature is present in a model trained on a subset of randomly sampled features from the full dataset versus when the feature is dropped from the sampled features and the model retrained. The size of the subset is often set to $\sqrt{k}$ where k is the number of variables; this was not effective for the smallest datasets. In this work, multiple subset sizes were examined and the one that best optimized the feature selection models was used. The third measure, last-in variable importance, represents the average change in balanced accuracy when the variable is added as the last variable to a model containing all other variables in the dataset.

The three measures indicate different properties of the variables. Variables with high first-in importance are useful as starting points when attempting to construct simpler models with fewer variables, as they already provide substantial predictive power by themselves. Sampled feature importance describes how variables contribute on average when included in a variety of different models, providing a more general sense of a variable's predictive power. Sampled importance is also relatively computationally expensive to calculate: it is important that variables are tested in a wide variety of random subsamples of the full variable set to provide the best importance estimates. Variables with high last-in importance provide contributions to a model's predictive power even in the presence of all other variables. This represents the predictive power unique to the variable not shared by the other variables.

The feature selection method used to train the machine learning models was sampled variable importance. Variable importance measures using first-in, last-in, and sampled variable importance are presented in Sec. III B.

III. RESULTS

A. Logistic regression

Table V presents the results of applying LR predicting earning a D or F in the class from all variables in each dataset (excluding the test average). All calculations were performed using the scikit-learn library in Python [33], the imbalanced-learn library for SMOTE upsampling [34], and the scikit-lego library for decision threshold tuning [35]. The Python implementation provides a richer and easier to customize implementation of a number of machine learning algorithms than those in "R." A series of LR-based models were used to attempt to improve prediction accuracy over an initial baseline LR model using default parameters. All models in Table V optimize the machine learning algorithm to maximize the balanced accuracy, $\bar{\alpha}$, when applied to the training dataset and are then applied to predict outcomes in the test dataset (which is not examined during training). Each model was fit 500 times on a dataset that sampled the original dataset with replacement (the data were bootstrapped). Because 500 replications were used, the standard error in the $\bar{\alpha}$ is the standard deviation divided by $\sqrt{500}$; as such, a 1% difference in $\bar{\alpha}$ is generally a significant difference. Both the train-test split and the bootstrapping were done such that the sample balance was preserved. For this study, a 70-30 train-test split was used.

The decision threshold is a model hyperparameter, a parameter external to the fitting process that governs model training. Grid-search cross validation as implemented in the Python scikit-learn library was used to select the decision threshold. This method is an extension of K-fold cross validation, which separates the training data into K equally sized subsamples or "folds" of the dataset. Each fold is then treated as a test dataset with the remaining folds used as a training set. This methodology helps to prevent overfitting and provides better generalizability to unseen data. Grid-search cross validation builds on this by constructing models with every possible permutation of hyperparameters provided (e.g., a list of potential decision thresholds) and identifying which hyperparameters optimize model performance for a given metric, in this case, the balanced accuracy. For this study, fivefold cross validation was used.

To evaluate the different classification methods, an overall range of balanced accuracy is helpful. In Table VII, we present our suggested variables for classification using our suggested algorithm and tuning; this model achieves a balanced accuracy of 82% on D_2 and D_3. For D_2, we then add the overall semester test average to the model. The test scores form 70% of the grade in the class studied and their average is easily the most valuable variable for the prediction of student grades. The addition of the test average increased balanced accuracy to only 89%. As such, there seems to be a ceiling on balanced accuracy for any reasonable set of variables.

Table V presents the results of several optimization techniques discussed in Sec. II applied to an LR classifier. Figure 1 provides a graphical representation of the same data for ease of comparison. The baseline model uses LR with the default settings that would be found in "R" with no regularization and with a decision threshold of $\eta = 0.5$. This was the model used in Zabriskie *et al.* [1]. As in that work, the classifier was excellent at correctly classifying passing students ($\alpha_{\mathrm{ABC}} = 0.97$), but fairly poor at correctly classifying failing students ($\alpha_{\mathrm{DF}} = 0.36$).

1. Decision threshold tuning

The decision threshold is the probability associated with a probabilistic classifier's ultimate classification: in our case, if the logistic regression yields a probability higher than the selected threshold for a student, that student is classified as at risk. Decision threshold tuning (DTT) changes the ABC and DF accuracy associated with a classifier. In this study, the threshold was selected to

TABLE V. Exploration of different optimizations of logistic regression models. $\bar{\alpha}_{\text{test}}$ is the balanced accuracy of the test dataset. $\bar{\alpha}_{\text{train}}$ is the balanced accuracy of the training dataset. α_{DF} is the success rate in predicting the DF students. α_{ABC} is the success rate in predicting ABC students. All are reported as mean ± standard deviation. Note that models in this table and all future tables were evaluated on 500 bootstrapped samples; as such, the standard errors are the standard deviation divided by $\sqrt{500}$ so that a 1% change is generally a significant difference. The test dataset balanced accuracy is bolded.

Algorithm	$\bar{\alpha}_{\text{test}}$	$\bar{\alpha}_{\text{train}}$	α_{DF}	α_{ABC}
		Dataset 0		
Baseline	$\mathbf{0.66 \pm 0.02}$	0.67 ± 0.01	0.36 ± 0.05	0.97 ± 0.01
Decision threshold tuning (DTT)	$\mathbf{0.78 \pm 0.02}$	0.79 ± 0.01	0.76 ± 0.06	0.79 ± 0.05
Regularization	$\mathbf{0.66 \pm 0.02}$	0.67 ± 0.01	0.36 ± 0.04	0.97 ± 0.01
SMOTE	$\mathbf{0.78 \pm 0.02}$	0.79 ± 0.01	0.73 ± 0.05	0.82 ± 0.02
DTT variable importance	$\mathbf{0.77 \pm 0.03}$	0.79 ± 0.02	0.76 ± 0.07	0.79 ± 0.05
DTT regularization	$\mathbf{0.78 \pm 0.02}$	0.79 ± 0.01	0.76 ± 0.05	0.81 ± 0.04
DTT SMOTE	$\mathbf{0.77 \pm 0.02}$	0.79 ± 0.01	0.75 ± 0.06	0.80 ± 0.05
Regularization SMOTE	$\mathbf{0.78 \pm 0.02}$	0.79 ± 0.01	0.73 ± 0.04	0.83 ± 0.02
		Dataset 1		
Baseline	$\mathbf{0.65 \pm 0.02}$	0.66 ± 0.01	0.33 ± 0.05	0.97 ± 0.01
Decision threshold tuning (DTT)	$\mathbf{0.78 \pm 0.03}$	0.80 ± 0.01	0.75 ± 0.08	0.81 ± 0.06
Regularization	$\mathbf{0.65 \pm 0.03}$	0.65 ± 0.01	0.32 ± 0.05	0.97 ± 0.01
SMOTE	$\mathbf{0.77 \pm 0.02}$	0.80 ± 0.01	0.73 ± 0.05	0.82 ± 0.02
DTT variable importance	$\mathbf{0.78 \pm 0.03}$	0.79 ± 0.02	0.76 ± 0.07	0.79 ± 0.06
DTT regularization	$\mathbf{0.78 \pm 0.02}$	0.79 ± 0.01	0.75 ± 0.06	0.81 ± 0.05
DTT SMOTE	$\mathbf{0.77 \pm 0.02}$	0.80 ± 0.01	0.74 ± 0.07	0.80 ± 0.05
Regularization SMOTE	$\mathbf{0.78 \pm 0.02}$	0.79 ± 0.01	0.73 ± 0.05	0.83 ± 0.02
		Dataset 2		
Baseline	$\mathbf{0.70 \pm 0.03}$	0.72 ± 0.02	0.43 ± 0.06	0.97 ± 0.01
Decision threshold tuning (DTT)	$\mathbf{0.82 \pm 0.02}$	0.84 ± 0.01	0.83 ± 0.07	0.81 ± 0.05
Regularization	$\mathbf{0.70 \pm 0.03}$	0.72 ± 0.02	0.42 ± 0.06	0.97 ± 0.01
SMOTE	$\mathbf{0.82 \pm 0.02}$	0.84 ± 0.01	0.79 ± 0.05	0.84 ± 0.02
DTT variable importance	$\mathbf{0.82 \pm 0.02}$	0.83 ± 0.01	0.83 ± 0.07	0.81 ± 0.05
DTT regularization	$\mathbf{0.82 \pm 0.02}$	0.84 ± 0.01	0.82 ± 0.06	0.82 ± 0.04
DTT SMOTE	$\mathbf{0.81 \pm 0.02}$	0.84 ± 0.01	0.82 ± 0.07	0.80 ± 0.05
Regularization SMOTE	$\mathbf{0.82 \pm 0.02}$	0.84 ± 0.01	0.80 ± 0.05	0.84 ± 0.02
		Dataset 3		
Baseline	$\mathbf{0.67 \pm 0.04}$	0.73 ± 0.02	0.36 ± 0.08	0.98 ± 0.01
Decision threshold tuning (DTT)	$\mathbf{0.78 \pm 0.03}$	0.84 ± 0.02	0.80 ± 0.10	0.77 ± 0.07
Regularization	$\mathbf{0.67 \pm 0.04}$	0.72 ± 0.02	0.36 ± 0.09	0.98 ± 0.01
SMOTE	$\mathbf{0.77 \pm 0.04}$	0.87 ± 0.02	0.69 ± 0.09	0.85 ± 0.02
DTT variable importance	$\mathbf{0.80 \pm 0.04}$	0.83 ± 0.02	0.80 ± 0.10	0.80 ± 0.07
DTT regularization	$\mathbf{0.81 \pm 0.03}$	0.83 ± 0.02	0.81 ± 0.08	0.82 ± 0.04
DTT SMOTE	$\mathbf{0.76 \pm 0.04}$	0.82 ± 0.04	0.84 ± 0.10	0.68 ± 0.12
Regularization SMOTE	$\mathbf{0.82 \pm 0.03}$	0.84 ± 0.02	0.81 ± 0.07	0.83 ± 0.02

optimize balanced accuracy which produced approximately equivalent performance on both at-risk and not-at-risk students. Table V shows that using DTT with LR (the decision threshold tuning entry in Table V) substantially improved balanced accuracy over the baseline model for all datasets. It also produced models with similar ABC and DF accuracy.

2. *Overfitting*

Examination of Table V shows that, for most models, the balanced accuracy of the test data predictions is approximately equal and slightly smaller than the balanced accuracy of the training data predictions. The training data predictions are expected to be better than the test

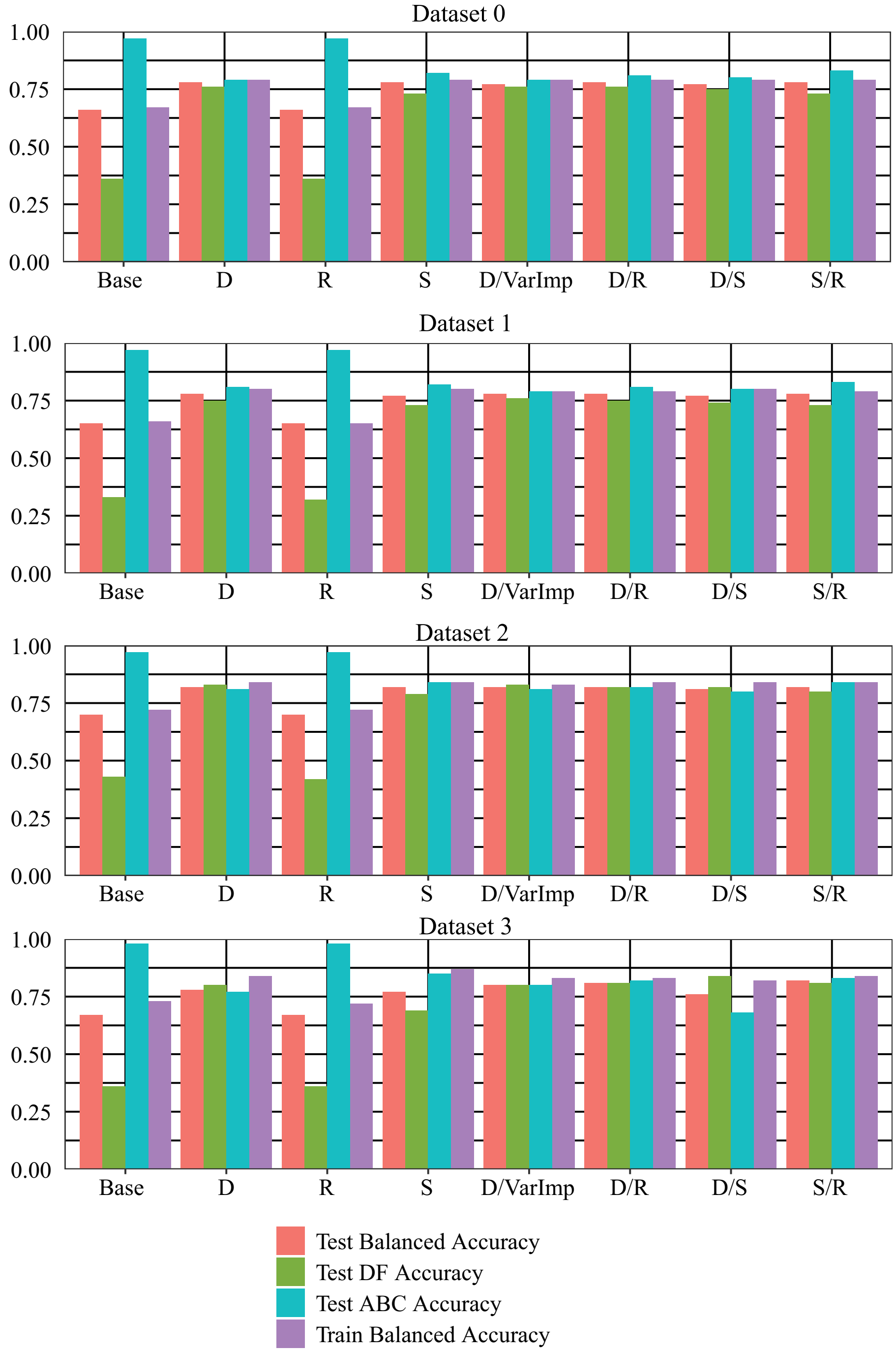

FIG. 1. The test and train balanced accuracy, test DF accuracy, and test ABC accuracy of each machine learning method on datasets 0 to 3. The figure presents a graphical representation of Table V. For reading in grayscale, each group of columns is in the same order as the legend at the bottom. “Base” represents baseline, “D” represents decision threshold tuning (DTT), “R” represents regularization, “S” represents SMOTE, “D/VarImp” represents DTT with variable importance, “D/R” represents DTT with regularization, “D/S” represents DTT with SMOTE, and “S/R” represents SMOTE with regularization.

predictions because the model was constructed on the training data while the test data represents completely new observations. For D_3, the model with DTT performs substantially better (6%) on the training dataset than on the test dataset. When performance on the training dataset substantially exceeds performance on the test dataset, the data are being “overfit,” the training model is being overly specialized to the training cases and ceases to perform as well on the new cases provided by the test dataset. This happens for D_3 because of the combination of a larger set of variables and a smaller number of students. The capability of detecting overfitting is a substantial strength of the train-test split methodology. Modeling methodologies that do not use a test-train split generally employ fit statistics which increase with additional parameters making the detection of overfitting difficult.

3. Feature selection

Both feature selection and regularization are meant to reduce overfitting and improve the generalizability of models to unseen data. Feature selection does this by including only a subset of variables from the full dataset that meet some criteria of inclusion. Regularization penalizes large regression coefficients during the fitting process to reduce model complexity, as complex models are more prone to overfitting.

The results of the feature selection analysis based on sampled feature importance, as discussed in Sec. II H, are shown in the DTT Variable Importance entries in Table V. A model was constructed with a subset of the available variables and the variable of interest; the variable of interest was then removed and the change in balanced accuracy recalculated. Only variables with an average change in balanced accuracy above some threshold were retained. Both the number of variables in the sample and the threshold for retaining variables were treated as hyperparameters; the hyperparameter space was searched for optimal combinations. Model performance for D_0 to D_2 did not change when feature selection was added to the algorithm using LR with DTT, but performance increased 2% for D_3 and the amount of overfitting was reduced by 50%. Implementing and tuning the variable importance models was challenging and computationally expensive; an alternate, computationally less intensive method to reduce overfitting would be helpful.

4. Regularization

Models using regularization without DTT performed approximately as well as the baseline model (entry regularization in Table V). Models using DTT and regularization (entry DTT regularization in Table V) performed equally to or outperformed those with DTT and variable importance (and require a fraction of the computational time to tune). These models also showed the same small amount of overfitting on D_3 that was observed in the smaller datasets.

5. SMOTE upsampling

SMOTE upsampling generates synthetic cases of the minority class by randomly selecting points on lines drawn between minority class records (in our case, it generates synthetic records for failing students). Performance differences between classes are common in imbalanced datasets; by removing this imbalance through generating a sufficient number of synthetic records, more equal performance on the two classes can be obtained.

Table V shows the results of using SMOTE to create training datasets with equal numbers of DF and ABC students. Models using SMOTE alone improved performance over the baseline model (entry SMOTE in Table V) by about the same amount as DTT improved the baseline model. DTT did not improve the performance of models also using SMOTE (entry DTT SMOTE in Table V); they both solve the same problem. Models using SMOTE and regularization (entry regularization SMOTE in Table V) were equal to or better than all other models. Prior PER works [1,2] likely failed to find SMOTE as an effective method of improving classification results because the metric optimized (overall accuracy, PPV) was sensitive to sample balance.

6. Alternate algorithms

One attractive feature of classification using machine learning is the extensive variety of algorithms offering the possibility of improving accuracy simply by using a different set of computer codes. The Supplemental Material [11] explores alternate algorithms including naive Bayes, random forests, support vector machines, and K-nearest neighbors using SMOTE to eliminate sample imbalance. Unfortunately, all produced models with similar predictive accuracy as LR at best. Likewise, ensemble models were also investigated where multiple different classifiers were constructed and allowed to vote on the classification. These also did not improve balanced accuracy above that of LR. While a fairly complex educational dataset, predicting student outcomes with the data used in this work does not require these additional algorithms.

7. Dataset performance differences

There are many differences between the four datasets including size and sample balance. The Supplemental Material [11] contains an investigation of the effects of these differences. Neither the difference in size nor the difference in sample balance affected prediction accuracy. The samples also contain somewhat different populations of students. For example, D_1 contains students in attendance when the pretest was given, while D_3 contains students willing to complete two optional surveys for bonus points. Dataset 3 was fit with the variables available in the other datasets; the balanced accuracy was 0.01 to 0.02 higher for D_0 to D_2 indicating the population in D_3 is slightly harder to predict.

8. Summary

In summary, LR using SMOTE with regularization produced the best-performing classifiers; however, their performance was equal to that of DTT with regularization for D_0 to D_2 and only improved performance by 1% for D_3.

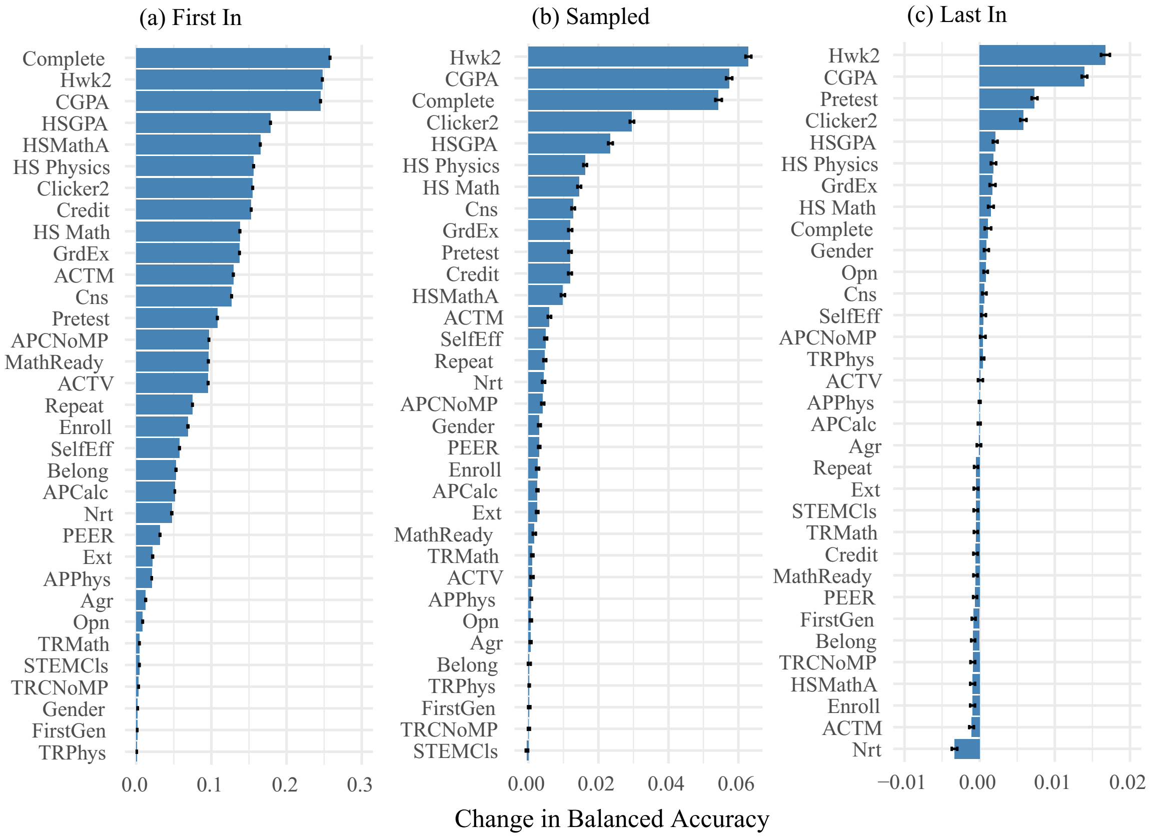

FIG. 2. The increase in balanced accuracy predicting dataset 3 when (a) First in—the variable is added to the null model containing only the intercept, (b) sampled—the variable is added to a randomly sampled set of variables, and (c) last in—the variable is added as the last variable in the full model. The length of the error bar is the standard error based on 500 replications.

As such, the intuitively simpler DTT with regularization may be the best choice for grade classification on datasets with a large number of variables. For D_0 to D_2, LR with DTT alone produced models with performance equal to the best models. Logistic regression with DTT is straightforward to implement in "R" (a working sample is provided in the Supplemental Material [11]) and may represent the best choice for instructors wishing to identify at-risk students.

B. Variable importance

Figure 2 presents the three measures of variable importance for D_3 which contained every available variable. Tables with variable importance measures for all datasets are provided in the Supplemental Material [11].

The variables Hwk2 and CGPA were of high importance, placing in the top three of all of the importance metrics. The percentage of attempted credit hours successfully completed for credit, Complete, was the highest first-in and the third highest sampled importance but had a moderate last-in importance. This suggests it is collinear with other variables, likely with CGPA. High school GPA and high school physics were also of relatively high importance on all metrics but not as important as the variables above. These variables had lower last-in importance than the variables above suggesting the majority of their predictive power was present in other variables. Grade expectation and conscientiousness were consistently the most predictive noncognitive variables with predictive power commensurate, but somewhat smaller, than the high school variables in both first-in and sampled importance; however, they had very little last-in importance. This again indicates that most of their predictive power is carried by other variables.

Note that the last-in predictive power of most variables was small. Several variables had negative last-in importances, suggesting they worsened model performance when included as the final variable. These variables contributed to overfitting when added as the last variable in the model.

The predictive power of groups of variables is described in Table VI. As might be expected from the individual variable importances, the in-class and college groups which include Hwk2 and CGPA were the most predictive, achieving a balanced accuracy of 0.77 on their own. General high school preparation, high school math, and high school physics also had substantial predictive power on their own (0.67, 0.66, and 0.64 respectively), but less power than the college and in-class variables. The student's expected grade was of equal power to their high school physics preparation. Noncognitive and demographic variables had little predictive power.

TABLE VI. Predictive power of variable groups predicting dataset 3. $\bar{\alpha}_{\text{test}}$ is the balanced accuracy of the test dataset. $\bar{\alpha}_{\text{train}}$ is the balanced accuracy of the training dataset. α_{DF} is the success rate in predicting DF students. α_{ABC} is the success rate in predicting ABC students. All are reported as mean ± standard deviation. The test dataset balanced accuracy is bolded.

Group	$\bar{\alpha}_{\text{test}}$	$\bar{\alpha}_{\text{train}}$	α_{DF}	α_{ABC}	Variables
Demographics	**0.51 ± 0.02**	0.53 ± 0.02	0.26 ± 0.33	0.77 ± 0.34	Gender, FirstGen, PEER
In-class	**0.77 ± 0.04**	0.77 ± 0.02	0.75 ± 0.09	0.79 ± 0.03	Hwk2, Clicker2
College	**0.77 ± 0.03**	0.78 ± 0.02	0.78 ± 0.09	0.75 ± 0.05	Complete, CGPA, STEMCls, Credit, Enroll
AP/transfer	**0.57 ± 0.03**	0.60 ± 0.02	0.81 ± 0.13	0.34 ± 0.10	APCNoMP, APPhys, APCalc TRCNoMP, TRPhys, TRMath
HS general	**0.67 ± 0.04**	0.69 ± 0.02	0.76 ± 0.13	0.58 ± 0.07	ACTM, ACTV, HSGPA, MathReady
HS physics	**0.64 ± 0.04**	0.66 ± 0.02	0.74 ± 0.14	0.54 ± 0.09	HSP.NAP.NA, HSP.NAP.A, HSP.APNP.NA HSP.APNP.A, HSP.APP.NA, HSP.APP.A
HS math	**0.66 ± 0.04**	0.68 ± 0.01	0.81 ± 0.14	0.52 ± 0.09	HSMathA, HSMNotAP, HSMAPNPass, HSMAPPass
Noncognitive	**0.59 ± 0.04**	0.63 ± 0.02	0.68 ± 0.19	0.51 ± 0.14	Belong, SelfEff, Agr, Cns, Nrt, Ext, Opn
Grade expectation	**0.64 ± 0.03**	0.64 ± 0.01	0.83 ± 0.06	0.44 ± 0.02	GradeExp

C. Suggested variable sets and algorithms

Figure 2 shows that the four most important variables using sampled variable importance were Hwk2, Clicker2, CGPA, and Complete. The first two are easily accessible for many physics instructors without additional effort. Beyond these, many classes give conceptual physics pretests; as such, the scores on these instruments are available without additional effort. Most institutions could provide CGPA upon request. Many other variables used in this work would require either a substantive interaction with institutional research or the deployment of optional survey instruments; both adding to the time burden on the instructor. The similarity of the performance of many models in Table V and the variable importance results of Sec. III B suggest that it may be possible to produce very good models with a limited set of variables. Table VII shows the results of fitting only what we believe are the most easy to obtain important variables. They will be called "convenient" variables. Comparison with Table V shows models including only Hwk2, Clicker2, Pretest, and CGPA perform as well as the full model for D_2 and D_3. Models containing only Hwk2 and Clicker2 perform only 1% less well than the best models for D_0 and D_1. Adding CGPA to the Hwk2 and Clicker2 models

TABLE VII. Predictive power of convenient variables. $\bar{\alpha}_{\text{test}}$ is the balanced accuracy of the test dataset. $\bar{\alpha}_{\text{train}}$ is the balanced accuracy of the training dataset. α_{DF} is the success rate in predicting DF students. α_{ABC} is the success rate in predicting ABC students. All are reported as mean ± standard deviation. The test dataset balanced accuracy is bolded.

$\bar{\alpha}_{\text{test}}$	$\bar{\alpha}_{\text{train}}$	α_{DF}	α_{ABC}	Variables
		Dataset 0		
0.76 ± 0.02	0.77 ± 0.01	0.75 ± 0.05	0.78 ± 0.03	Hwk2, Clicker2
		Dataset 1		
0.77 ± 0.02	0.77 ± 0.01	0.76 ± 0.06	0.78 ± 0.03	Hwk2, Clicker2
0.77 ± 0.02	0.78 ± 0.01	0.76 ± 0.06	0.78 ± 0.04	Hwk2, Clicker2, Pretest
		Dataset 2		
0.77 ± 0.03	0.78 ± 0.01	0.74 ± 0.07	0.80 ± 0.03	Hwk2, Clicker2
0.77 ± 0.03	0.78 ± 0.01	0.73 ± 0.07	0.82 ± 0.04	Hwk2, Clicker2, Pretest
0.80 ± 0.03	0.81 ± 0.01	0.81 ± 0.08	0.80 ± 0.05	Hwk2, Clicker2, CGPA
0.82 ± 0.02	0.83 ± 0.01	0.83 ± 0.06	0.81 ± 0.04	Hwk2, Clicker2, Pretest, CGPA
0.89 ± 0.02	0.90 ± 0.01	0.90 ± 0.06	0.87 ± 0.03	Hwk2, Clicker2, Pretest, CGPA, TestAve
		Dataset 3		
0.77 ± 0.04	0.77 ± 0.02	0.75 ± 0.09	0.79 ± 0.03	Hwk2, Clicker2
0.76 ± 0.04	0.77 ± 0.02	0.69 ± 0.10	0.83 ± 0.06	Hwk2, Clicker2, Pretest
0.80 ± 0.03	0.81 ± 0.01	0.83 ± 0.08	0.80 ± 0.04	Hwk2, Clicker2, CGPA
0.82 ± 0.03	0.83 ± 0.02	0.84 ± 0.09	0.80 ± 0.05	Hwk2, Clicker2, Pretest, CGPA
0.81 ± 0.03	0.83 ± 0.01	0.84 ± 0.09	0.79 ± 0.05	Hwk2, Clicker2, Pretest, CGPA, GradeExp

improved balanced accuracy by 3% for D_2 and 4% for D_3. Examination of Table VII shows that the addition of the pretest to Hwk2 and Clicker2 does not increase balanced accuracy for any dataset; however, the addition of pretest to Hwk2, Clicker2, and CGPA increases balanced accuracy by 2% in both D_2 and D_3. With this restricted set of variables, regularization is no longer needed. The Supplemental Material [11] refits the results of Table VII without regularization; the balanced accuracy of all models was identical.

As such, an instructor wishing to implement a classification algorithm for their students using a small set of fairly easy to obtain variables can use the in-class variables they have available (in this case, Hwk2 and Clicker2) with logistic regression using DTT (a working code sample in “R” is presented in the Supplemental Material [11]). Instructors wishing to improve classification accuracy by 5% to 6% could request the student’s CGPA from institutional records. With CGPA, the addition of pretest scores can increase balanced accuracy by an additional 2%. The results above suggest these classifiers perform nearly as well as those using much larger sets of variables and more sophisticated algorithms.

IV. DISCUSSION

A. Research questions

This work investigated three research questions. These have been discussed in detail in the previous section and will be summarized below.

RQ1: How can machine learning outcomes be optimized to most effectively predict student outcomes early in physics classes? By employing techniques to account for the large sample imbalance between students who pass and fail an introductory mechanics course, models using a combination of in-class, and institutional data available by the second week of the course were constructed. Both tuning the classification decision threshold to optimize balanced accuracy and using SMOTE upsampling yielded the best-performing models with a balanced accuracy of 82% for D_2. The model using DTT had almost identical performance on both passing and failing students, with a DF accuracy of 83% and an ABC accuracy of 81%. The SMOTE model had a slightly larger performance gap between the two classes, with a DF accuracy of 79% and an ABC accuracy of 84%. Both represent a substantially more balanced performance on the two outcomes relative to a pure guessing model, as well as the baseline logistic regression model. The baseline model, which made no attempt to correct the sample imbalance, had a DF accuracy of 43% and an ABC accuracy of 97%. Regularization and feature selection methods helped prevent overfitting of the training data and improved the generalizability of the model to test data, but these effects were only substantial on dataset 3, which had both the smallest number of students and the largest number of variables included.

Machine learning algorithms other than LR were explored; these results are described in the Supplemental Material [11]. Support vector machine and random forest classifiers achieved comparable balanced accuracy to LR, but generally took longer to train. Other methods produced lower balanced accuracy than LR. Ensemble methods where multiple classifiers are built and vote on the classification were also investigated but performed markedly worse than the LR models. As such, LR using SMOTE upsampling produced the most accurate results. LR with DTT and regularization produced equivalent results except for dataset 3 where the balanced accuracy for the test dataset was 1% lower. These models produced significantly higher DF accuracy than prior PER studies [1,2].

RQ2: How does the performance of the algorithms change with the addition of new types of variables such as noncognitive or institutional variables? What factors are most important in the prediction of student success in physics classes? In general, the addition of both a richer set of high school variables and noncognitive variables did not improve prediction accuracy. This was partially the result of students in dataset 3 being somewhat more difficult to predict (1% to 2%) as shown in the Supplemental Material [11]. Three variable importance metrics were used to evaluate the importance of each variable to the model’s classification performance: first-in importance, sampled importance, and last-in importance. Results from all three suggest that both institutional and in-class variables are of high importance to the model’s classification results; however, only a few institutional variables were of high importance. Noncognitive variables such as students’ grade expectations, self-efficacy, sense of belonging, and personality were less important to the models and did not provide substantial predictive power. The in-class variables corresponding to the average second-week homework and clicker participation scores and the student’s CGPA were consistently of high importance across multiple variable importance metrics.

RQ3: How does the performance of the optimized model compare with that using a limited set of variables easily accessible to physics instructors? Models trained on only a subset of the available features had comparable performance to the model trained on the full dataset. As shown in Table VII, on D_2, a balanced accuracy of 82% was obtained using a model trained only on the second week average homework and clicker scores, CGPA, and FMCE pretest scores. This was equivalent to the performance of the model trained on all variables available in D_2 and uses common variables obtainable by the second week through in-class grade collection and an institutional data request. A model provided with only second week average homework and clicker scores achieved a balanced accuracy of 77% on D_0 to D_3, a marked increase over pure guessing without any use of variables external to the course itself. These results suggest that strong predictive models can be constructed even from relatively small sets of features. Table VI supports this observation showing the in-class and institutional variable sets with equal predictive power and much higher predictive power than other groups.

This further supports the conclusion that the collection of extensive institutional datasets or augmenting in-class and institutional data with additional survey data may not lead to improved prediction accuracy.

B. Other observations

Machine learning contributes an important new conceptual technology to the exploration of dichotomous outcomes, the idea of classification. A LR model intrinsically assigns a probability of passing to each student. By applying a decision threshold, this probability is turned into a classification prediction. Using the confusion matrix allows the computation of a rich suite of statistics to characterize many features of this prediction: not only how often the prediction is right but also the features of incorrect predictions. Are you more likely to predict an eventually failing student will succeed or an eventually successful student will fail? These additional metrics beyond the simple prediction of the probability of success should allow for a more nuanced exploration of the features that make certain students difficult to predict and the features that cause the classifier to make certain kinds of errors.

One primary methodology in machine learning, the division of data into a training and test dataset, would also be generally useful in PER. This methodology allowed the detection of overfitting; something that is likely a common problem in PER, but which is often not explored.

Machine learning methods also provide new ways to characterize variable importance; both sampling the variables and examining the change in balanced accuracy (or one of many other classification metrics) provided a promising technique to reduce multicollinearity and provided a much more intuitive measure of variable importance. Traditional measures of variable importance examine the additional variance explained or the size of the statistics used to characterize significance (such as t); the increase in the balanced accuracy when the variable is added to the model is far more intuitive and gives a measure of the practical importance of the addition of a variable to the model. The test-train split methodology also allowed for variables with negative importance; variables which made the models worse when added. This provides another way to detect overfitting.

C. Ethical considerations

Classification predictions are a tool to help instructors improve learning outcomes; they can be properly used or misused. Instructors must take care to ensure that model results do not bias their treatment of individual students. No model is 100% accurate; students identified as being at risk of failure may earn a passing grade in the course; those not identified as at risk may fail. Additionally, results should not be used to exclude students classified as not at risk from additional instructional resources: changes made to a course's structure to benefit at-risk students should also be made available to all students. Instructors should also be aware that the accuracy of predictive models is sensitive to a particular course's instructional conditions [36] and should be cognizant of broader ethical considerations regarding the use of institutional data [37]. In particular, care should be taken to examine how model performance differs across different demographic subgroups in the data. A model trained on data consisting primarily of students from a majority group may display performance differences in its classification of students from minoritized groups.

The Supplemental Material [11] presents an analysis exploring classification accuracy for women, PEER, and FGCS students. Classification models were trained on all students; these models were then used to predict women, PEER students, and FGCS students. Generally, the success rate for predicting D or F grades, α_{DF}, was similar for the majority of students and other students; however, the α_{DF} was somewhat lower for women while α_{ABC} was higher. Conversely, α_{DF} was higher for PEER students and α_{ABC} lower. Performance on FGCS students was close to the overall performance on D_2 and D_3. This suggests the decision threshold should be somewhat retuned when predicting women and PEER students. In general, model accuracy was not improved by building the classifier on only students from the demographic subgroups. When possible, instructors using classification should confirm the accuracy for all groups of interest.

D. Future—Using classification in the classroom

Classification has many potential uses in the classroom. Classification could be used as part of a class early warning system that provides students with a visual indication of their status [14]. The classification could be used to direct additional monitoring of or messaging to at-risk students. The messaging could take the form of general advice on how to get additional help in the class or reminders of class policy such as the acceptance of homework after the due date for a small penalty. The classification could also be used to organize subgroups such as lab groups within the class. Additional research would be needed to determine how to do so effectively. In general, a student's risk classification represents another variable that instructors and researchers can use to improve student outcomes. Additional research would be needed to determine which of these possible methods are most effective and to determine how to best apply each method.

V. LIMITATIONS

This study was performed using data from a single course at a single large research university with a majority White male enrollment. The results should be replicated in many other classes at different levels at different institutions to determine if the results are generalizable. It is particularly important that this be done for institutions with student populations with a different demographic composition and for institutions other than research universities.

The methods achieved substantial predictive accuracy but not perfect accuracy. The features of students incorrectly classified by the optimal models should be investigated to identify additional information that could be collected to further improve prediction accuracy.

VI. CONCLUSIONS

Machine learning models were constructed to predict if a student was at risk or not at risk of failing an introductory mechanics course. Models were trained on a combination of in-class, institutional, and noncognitive variables available by the second week of class; early prediction was prioritized so that student interventions might be implemented early in the semester. Logistic regression models combined with either decision threshold tuning or SMOTE upsampling yielded the best results overall, with nearly equal performance on both passing and failing students. Variable importance metrics suggest that students' average homework scores and their college GPAs were among the variables most important to the models' decision making. A logistic regression classifier with decision threshold tuning using homework grades, clicker scores, and college GPA had performance near models using an extensive set of variables and may be the best choice for in-class prediction. Various other machine learning algorithms and techniques were not effective at improving model performance over logistic regression.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under Grants No. ECR-1561517, No. HRD-1834569, and No. DUE-1833694.

[1] C. Zabriskie, J. Yang, S. DeVore, and J. Stewart, Using machine learning to predict physics course outcomes, Phys. Rev. Phys. Educ. Res. **15**, 020120 (2019).

[2] J. Yang, S. DeVore, D. Hewagallage, P. Miller, Q. X. Ryan, and J. Stewart, Using machine learning to identify the most at-risk students in physics classes, Phys. Rev. Phys. Educ. Res. **16**, 020130 (2020).

[3] J. M. Aiken, R. Henderson, and M. D. Caballero, Modeling student pathways in a physics bachelor's degree program, Phys. Rev. Phys. Educ. Res. **15**, 010128 (2019).

[4] President's Council of Advisors on Science and Technology, *Report to the President. Engage to Excel: Producing One Million Additional College Graduates with Degrees in Science, Technology, Engineering, and Mathematics* (Executive Office of the President, Washington, DC, 2012).

[5] C. Romero and S. Ventura, Educational data mining and learning analytics: An updated survey, WIREs Data Min. Knowl. **10**, e1355 (2020).

[6] A. Peña-Ayala, Educational data mining: A survey and a data mining-based analysis of recent works, Expert Syst. Appl. **41**, 1432 (2014).

[7] H. Aldowah, H. Al-Samarraie, and W. Fauzy, Educational data mining and learning analytics for 21st century higher education: A review and synthesis, Telemat. Inform. **37**, 13 (2019).

[8] C. Romero, S. Ventura, P. G. Espejo, and C. Hervás, Data mining algorithms to classify students, in *Proceeding of the 1st International Conference on Educational Data Mining*, edited by R. S. J. de Baker, T. Barnes, and J. E. Beck (International Educational Data Mining Society, Boston, MA, 2008), https://www.educationaldatamining.org/EDM2008/proceedings.html.

[9] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Applications in R* (Springer-Verlag, New York, NY, 2017), Vol. 112.

[10] A. C. Müller and S. Guido, *Introduction to Machine Learning with Python: A Guide for Data Scientists* (O'Reilly Media, Boston, MA, 2016).

[11] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevPhysEducRes.20.010149 for fairness analysis, variable importance for each dataset, data subsetting analysis, and sample "*R*" classification code.

[12] R. Alkhasawneh and R. H. Hargraves, Developing a hybrid model to predict student first year retention in STEM disciplines using machine learning techniques, J. STEM Educ. **15**, 35 (2021), https://www.jstem.org/jstem/index.php/JSTEM/article/view/1805.

[13] Ma. V. Q. Almeda and R. S. Baker, Predicting student participation in STEM careers: The role of affect and engagement during middle school, J. Educ. Data Min. **12**, 33 (2020).

[14] A. Cano and J. D. Leonard, Interpretable multiview early warning system adapted to underrepresented student populations, IEEE Trans. Learn. Technol. **12**, 198 (2019).

[15] C. P. Rosé, E. A. McLaughlin, R. Liu, and K. R. Koedinger, Explanatory learner models: Why machine learning (alone) is not the answer, Br. J. Educ. Technol. **50**, 2943 (2019).

[16] D. Spikol, E. Ruffaldi, L. Landolfi, and M. Cukurova, Estimation of success in collaborative learning based on multimodal learning analytics features, in *Proceedings of the 2017 IEEE 17th International Conference on Advanced Learning Technologies, ICALT, Timisoara, Romania* (IEEE, New York, 2017), pp. 269–273.

[17] A. Bogarín, R. Cerezo, and C. Romero, Discovering learning processes using inductive miner: A case study with learning management systems (LMSs), Psicothema **30**, 322 (2018).

[18] J. L. Rastrollo-Guerrero, J. A. Gómez-Pulido, and A. Durán-Domínguez, Analyzing and predicting students' performance by means of machine learning: A review, Appl. Sci. **10**, 1042 (2020).

[19] S. Kotsiantis, C. Pierrakeas, and P. Pintelas, Predicting students' performance in distance learning using machine learning techniques, Appl. Artif. Intell. **18**, 411 (2004).

[20] S. Huang and N. Fang, Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models, Comput. Educ. **61**, 133 (2013).

[21] S. D. A. Bujang, A. Selamat, O. Krejcar, F. Mohamed, L. K. Cheng, P. C. Chiu, and H. Fujita, Imbalanced classification methods for student grade prediction: A systematic literature review, IEEE Access **11**, 1970 (2023).

[22] National Center for Education Statistics, https://nces.ed.gov/collegenavigator [accessed September 7, 2023].

[23] E. Mazur, *Peer Instruction: A User's Manual* (Prentice Hall, Upper Saddle River, NJ, 1997).

[24] R. K. Thornton and D. R. Sokoloff, Assessing student learning of Newton's laws: The Force and Motion Conceptual Evaluation and the evaluation of active learning laboratory and lecture curricula, Am. J. Phys. **66**, 338 (1998).

[25] R. K. Thornton, D. Kuhl, K. Cummings, and J. Marx, Comparing the force and motion conceptual evaluation and the Force Concept Inventory, Phys. Rev. ST Phys. Educ. Res. **5**, 010105 (2009).

[26] D. J. Asai, Race matters, Cell **181**, 754 (2020).

[27] P. R. Pintrich, D. A. F. Smith, T. Garcia, and W. J. Mckeachie, Reliability and predictive validity of the Motivated Strategies for Learning Questionnaire (MSLQ), Educ. Psychol. Meas. **53**, 801 (1993).

[28] C. Good, A. Rattan, and C. S. Dweck, Why do women opt out? Sense of belonging and women's representation in mathematics., J. Pers. Soc. Psychol. **102**, 700 (2012).

[29] L. R. Goldberg, The development of markers for the big-five factor structure, Psychol. Assess. **4**, 26 (1992).

[30] O. P. John, E. M. Donahue, and R. L. Kentle, *The Big Five Inventory–Versions 4a and 54* (Institute of Personality and Social Research, University of California Berkeley, Berkeley, CA, 1991).

[31] O. P. John, L. P. Naumann, and C. J. Soto, Paradigm shift to the integrative big five trait taxonomy: History, measurement, and conceptual issues., in *Handbook of Personality: Theory and Research* (The Guilford Press, New York, NY, 2008), p. 114.

[32] M. Richardson, C. Abraham, and R. Bond, Psychological correlates of university students' academic performance: A systematic review and meta-analysis., Psychol. Bull. **138**, 353 (2012).

[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. **12**, 2825 (2011), https://www.jmlr.org/papers/v12/pedregosa11a.html.

[34] G. Lemaître, F. Nogueira, and C. K. Aridas, Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning, J. Mach. Learn. Res. **18**, 1 (2017), https://www.jmlr.org/papers/v18/16-365.html.

[35] scikit-lego (2019), available at https://scikit-lego.netlify.app/.

[36] D. Gasevic, S. Dawson, T. Rogers, and D. Gasevic, Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success, Internet Higher Educ. **28**, 68 (2016).

[37] L. D. Roberts, V. Chang, and D. Gibson, Ethical considerations in adopting a university- and system-wide approach to data and learning analytics, in *Big Data and Learning Analytics in Higher Education: Current Theory and Practice*, edited by B. K. Daniel (Springer International Publishing, New York, 2017), pp. 89–108.