The application of predictive analytics to identify at-risk students in health professions education

Short title: Predictive analytics to identify at-risk health professions students

August 4, 2021

Authors

Anshul Kumar
MGH Institute of Health Professions
akumar@mghihp.edu
(first and corresponding author)

Roger A. Edwards
MGH Institute of Health Professions
(co-senior author)

Lisa Walker
MGH Institute of Health Professions
(co-senior author)

This document contains the following items, within a single file:

1. Article manuscript
2. Supplementary code and data appendix

Abstract

Introduction: When a learner fails to reach a milestone, educators often wonder if there had been any warning signs that could have allowed them to intervene sooner. Machine learning is used to predict which students are at risk of failing a national certifying exam. Predictions are made well in advance of the exam, such that educators can meaningfully intervene before students take the exam.

Methods: Using already-collected, first-year student assessment data from four cohorts in a Master of Physician Assistant Studies program, the authors implement an "adaptive minimum match" version of the k-nearest neighbors algorithm (AMMKNN), using changing numbers of neighbors to predict each student's future exam scores on the Physician Assistant National Certifying Examination (PANCE). Leave-one-out cross validation (LOOCV) was used to evaluate the practical capabilities of this model, before making predictions for new students.

Results: The best predictive model has an accuracy of 93%, sensitivity of 69%, and specificity of 94%. It generates a predicted PANCE score for each student, one year before they are scheduled to take the exam. Students can then be prospectively categorized into groups that need extra support, optional extra support, or no extra support. The educator then has one year to provide the appropriate customized support to each type of student.

Conclusions: Predictive analytics can help health professions educators allocate scarce time and resources across their students. Interprofessional educators can use the included methods and code to generate predicted test outcomes for students. The authors recommend that educators using this or similar predictive methods act responsibly and transparently.

Introduction

Students in health professions programs must typically pass a certification exam before they can practice in their field. Obviously, it is the goal of a) students to pass the exam on their first attempt and b) educators to have tools to maximize student success on exams. We use predictive analytics and machine learning methods to develop one such tool and demonstrate how it can be applied within a health professions education program. *The purpose of this tool is to predict (guess) each student's score on a national certification exam one year*

before the student actually takes the exam. These predicted scores—if they are accurate enough—can help identify students who might benefit from additional support from educators before the exam.

We use machine learning to look for patterns in the data of previous cohorts of students, whose certification exam scores are known. We then use these established patterns to make predictions about the future certification exam scores of currently enrolled students. Our data come from four now-graduated cohorts of students in a Master of Physician Assistant Studies program. Such predictive analytic approaches are used in a variety of educational settings (Kučak et al. 2018). As Ekowo and Palmer (2016) describe, these approaches can be used to both discriminate against or uplift and support vulnerable students, depending on the intentions of the user.

Existing scholarship tends to a) identify at-risk students using standard available machine learning approaches, and b) focus more on model-building than on how to practically leverage predictive modeling when working with students in a health professions program. We build on this work by a) developing and applying a new type of "adaptive minimum match" k-nearest neighbors (AMMKNN) algorithm that uses different numbers of neighbors for each prediction and prioritizes identification of at-risk students—the educator's true goal, we argue—over predictive accuracy, and b) proposing how analytics results can be practically and ethically applied within the context of a health professions education program. Additionally, we have published the R code used for our analysis in an open source R package, called AdaptiveLearnalytics (Kumar and Edwards 2021), for others to use our methods.

Background

Predictive analytics in education

A typical approach to educational analytics is to use student performance or behavior data at intermediate stages to predict a final outcome of interest. For example, student results on homework assignments or quizzes as well as student activity records in online learning management systems during a semester-long course can be used to make guesses about final exam performance. This has been demonstrated with students studying a number of topics, including informatics (Kotsiantis 2012), human computer interaction (Stimpson et al. 2014), computer hardware (Akçapınar et al. 2019), and mathematics (Sokkhey and Okazaki 2019). Liz-Domınguez et al. (2019) refer to this as *grade prediction*. The same approach has also been used to predict which students will drop out of an educational

program (Aulck et al. 2016; Waheed et al. 2019). These studies train and evaluate the usefulness of machine learning models using a number of metrics such as accuracy, sensitivity, specificity, and AUC (area under the curve). Such machine learning models can be used on future groups of students to create early warning systems that identify at-risk students and intervene, as described by Arnold and Pistilli (2012).

Within health professions education specifically, Black et al (2021) have taken a similar approach within the physician assistant education context. Predictive analytic methods have been used on data from students learning oral pathology (Walkowski et al. 2015), blended medicine (Saqr et al. 2017), and psychomotor skills (Chan et al. 2020). It has also been used to evaluate surgical competence (Bissonnette et al. 2019; Winkler-Schwartz et al. 2019) and essays written by medical students (Gierl et al. 2014). Chan and Zary (2019), Tolsgaard et al. (2020), and ten Cate et al. (2020) review studies related to analytics in health professions education. All of these examples show that predictive techniques can be used to help educators make guesses about future student outcomes.

Physician assistant education and team-based learning (TBL)

The physician assistant (PA) profession is currently undergoing tremendous growth in the United States. To become a certified PA in the United States, students typically study for two or more years and then must achieve a score of at least 350 on the Physician Assistant National Certifying Exam (PANCE; graded on a 200–800 scale). In a typical PA program, the first year of education primarily features coursework with regular assessments and the second year consists of clinical rotations. Given these characteristics, PA programs are a good example of a health professions program that might benefit from predictive analytic tools.

Our data come from a PA program that mostly uses team-based learning (TBL), which is relatively new in PA education and has been growing in popularity in medical and other health professions programs (Wallace and Walker 2018). TBL relies on small group interaction, creating opportunities for students to solve problems together and practice key concepts (Michaelsen and Sweet 2008). The learning process consists of three stages: student preparation, readiness assurance, and application (McMahon 2010). Wallace and Walker (2018) describe the following key details of TBL: a) Create TBL assessments and learning activities using backwards design, b) Form diverse teams of 5–7 students, c) Students study content before class. In class, they engage in individual and team assessments, listen to minilectures, and participate in application activities. TBL

assessments include iRATs (individual readiness assurance test) and tRATs (team readiness assurance test). Students in this PA program also take the PACKRAT (Physician Assistant Clinical Knowledge Rating and Assessment Tool) exam at the end of their first year in the program. The PACKRAT is a cumulative, nationally-benchmarked assessment that covers important PA knowledge areas.

The frequency of assessments in TBL curricula generates data that can be easily included in predictive models. However, such an abundance of data is by no means necessary for using our methods. In fact, we ended up combining or eliminating many pieces of information. Health professions education programs that do not generate as much data can still use predictive analytics.

Research Methods

Our study uses data on student performance from a Master of Physician Assistant (PA) Studies program to create a predictive machine learning model to identify students at potential risk of failing the Physician Assistant National Certifying Exam (PANCE). This research was approved by the Mass General Brigham institutional review board in 2020 (protocol 2020P000514). Table 1 shows the process of data collection, analysis, and application of results as we plan to use it in practice. We used R (R Core Team 2020) for all analysis. We also developed an R package called AdaptiveLearnalytics (Kumar and Edwards 2021) to carry out key portions of our analysis. Our entire analysis and details on using the new package can be found in supplemental file *S1 code and data appendix*.

Table 1: Entire two-year analytics process from program perspective

Time →	Pre-matriculation	Year 1	End Year 1	Year 2	End Year 2
Process step →	Gather data: Undergraduate GPA	Gather data: iRAT quizzes, final exams, final course grades, PACKRAT score	Run machine learning model to identify at-risk students	Provide additional support to predicted at-risk students.	All students take National Certifying Exam

Data description and preprocessing

We utilized de-identified student data from four successive cohorts starting in 2015—a total of 180 students—from a single PA program. The data set was organized such that each row is a student and each column is a variable (containing a measured grade or other characteristic). We utilize iRAT quiz, final exam, final course grades, PACKRAT scores from

students' first year in the PA program, and undergraduate grade point average (GPA) data. Even though regular assessments do occur throughout the second year of the PA program, we do not use this data or any demographic data. We exclude second-year data so that an entire year is available to provide additional support to students who are at risk of failing the PANCE.

Our dependent variable is the PANCE score from each student's first attempt at this exam, after the student completes the 25-month PA program. PANCE scores can range from 200–800 points, and students must score above 350 to pass. We use student performance variables in the data set, described above, as independent variables to predict students' PANCE scores.

This data set, collected as part of this PA program's mostly-TBL curriculum, gives us more information about each student than we would expect to have in a traditional (non-TBL) curriculum. For each TBL course, we calculated the mean of each student's iRAT scores and eliminated the individual iRAT score variables. For example, if students completed seven iRAT quizzes during Course A, we made a new variable with the mean of those seven iRAT quiz scores. We then eliminated the seven individual iRAT quiz scores from our data set.

After these preprocessing steps, we had the following variables for each student for 19 first year courses: iRAT mean (when applicable), final exam score, final grade. We also had the following additional variables: overall undergraduate GPA, undergraduate science GPA, first-year PACKRAT score.

Once this manual preprocessing of variables was complete, we further narrowed down our independent variables by calculating the Pearson correlation coefficient of each independent variable with PANCE score. Any variable correlated with PANCE at less than 0.2 was eliminated. This correlation threshold was found to select the set of variables that led to the best model predictions. After these steps, a total of 54 variables remained that were included for possible use in a predictive machine learning model. The full list of variables can be found in the file *S1 code and data appendix*.

Model selection and adaptive minimum match KNN (AMMKNN)

Since a numeric score is being used to determine if a student passes or fails the PANCE, ours is both a classification and regression problem. It is classification because we are trying to predict if a student passes or fails. It is regression because test results are numeric scores. As educators, we decided it would be most useful to classify students into three groups: very likely to fail (predicted PANCE score <350), moderate risk of failing

(predicted PANCE score 350–375), low risk of failing (predicted PANCE score >375). The "moderate risk" category ensures that we were able to identify "at risk" students and not just those who would fail. This additional category supports our goal of using predictive analytics to help students advance toward examination with the best preparation resources that we can provide.

Before settling on a single predictive technique, we started by using a number of standard approaches, such as random forest (RF), support vector machine (SVM), and standard K-nearest neighbors (KNN). These attempts are presented in the supplementary file S1. None of these allowed us to create a predictive model that would achieve our educational goals of prioritizing the detection of students who might fail the PANCE while still keeping incorrect predictions within tolerable limits.

We developed a modified version of KNN called "adaptive minimum match" KNN (AMMKNN), summarized below. If we imagine a student named X who has just finished their first year of the PA program, this is how we predict X's PANCE score:

1. Start with a *training* data set of all previously-graduated students from the program, in which rows are students and columns are only the 54 selected first-year independent variables (the dependent variable, PANCE score, is omitted).
2. Rank the training students from closest to X to farthest from X, based on Euclidean distance.
3. Select the 20 closest matches to X out of the ranked training students. This gives us a list of X's 20 nearest neighbors, to be used in subsequent steps. So far, this procedure is the same as standard KNN with K = 20.
4. We will now use the (known) PANCE scores of these 20 closest matches to make a guess about X's (unknown) future PANCE score.
5. Calculate the mean PANCE score of X's 1 closest neighbor (which is just that neighbor's score itself). Calculate the mean PANCE score of X's 2 closest neighbors. Calculate the mean PANCE score of X's 3 closest neighbors. Continue this process until there are 20 means, each calculated on a different number of X's nearest neighbors. Find the lowest out of the 20 calculated means, called X's *minimum of means*.
6. Identify the lowest PANCE score out of X's 20 nearest neighbors. This is X's *minimum match*.
7. Determine if X's PACKRAT (the most important independent variable) score is less or greater than 2 standard deviations below the mean PACKRAT score. (This is to flag students who are possible outliers on the lower end).

a. If greater (which is common): X's predicted PANCE score is their minimum of means.
b. If less (which is rare): X's predicted PANCE score is their minimum match.

We can do this procedure each year for Student X and all of their classmates who are in between years 1 and 2 of the PA program. The training data set contains all students from previous cohorts who have completed their first attempt of the PANCE (after they have graduated from the PA program). Standard KNN uses the same value of K—the number of matches—for each student's score prediction. It would then treat the mean of those first K matches as the predicted value for every student. Our adaptive approach differs because it uses a different value of K for each prediction. In the example above, our approach uses a different value of K—based on either the minimum of means or minimum match—to make a prediction for Student X and each of their classmates. Other adaptive KNN procedures have been used in different contexts (Ougiaroglou et al. 2007; Sun and Huang 2010; Kibanov et al. 2018).

The maximum possible value of K was set at 20 in our study. This means that anywhere between 1 and 20 nearest neighbors could be used to make a prediction. We varied this maximum number of K and found that numbers as low as 10 and as high as 30 also appear to produce similar results. The consequence of our "adaptive minimum match" modification to KNN is that our predicted PANCE scores will be lower than they would be with the standard KNN algorithm (and many other commonly-used predictive models). This decision was deliberately made to prioritize the detection of students who might fail the PANCE rather than maximizing overall accuracy of the predictive model.

Model evaluation

We evaluated the results of all predictive models using leave-one-out cross validation (LOOCV), which has been used or recommended before in similar situations with small sample sizes (Zohair 2019; Black et al. 2021). We adhered as closely as possible to recommendations from Rao et al. (2008). To execute LOOCV on our sample of 180 students with a given predictive approach, we trained the model on 179 students and tested it on the remaining one student. We repeated this 180 times, such that each student was the testing student one time. This gives us a predicted numeric PANCE score for each student that the computer calculated without "knowing" the true PANCE score of that student. We then classify students as predicted to pass (if their predicted score is greater than 350) or fail (lower than 350).

For each predictive model, a confusion matrix was then created to show each student's predicted value (from when they were the testing student) and their actual PANCE result. From the confusion matrices for each model, we calculate the following standard metrics:

- True positives (TP)- The number of students who truly failed PANCE and were correctly predicted by the model to fail PANCE. To us as educators, these are "predictable support" students who truly need additional remedial support and our model succeeded in identifying them as such.
- False positives (FP)- The number of students who truly passed PANCE but were incorrectly predicted to fail. These are "unnecessary support" students who do not truly need our support but our model tells us that they do need remediation.
- True negatives (TN)- The number of students who truly passed PANCE and were correctly predicted to pass. These "predictable no support" students did not need our support and the model correctly predicted this.
- False negatives (FN)- The number of students who truly failed PANCE but were incorrectly predicted to pass. These students "fell through the cracks" because the model would ideally identify them as needing remedial support, but it did not. We argue that it is our duty to reduce this number as much as possible.
- Accuracy- The total number of correct predictions divided by the total number of students, (TP+TN)/180.
- Sensitivity- The proportion of students who truly failed who were correctly predicted to fail, TP/(TP+FN).
- Specificity- The proportion of students who truly passed who were correctly predicted to pass, TN/(TN+FP).

We define "positive" and "negative" outcomes this way to maintain consistency with the use of these terms in healthcare: A positive outcome is unwanted, like failing a test or having a disease; a negative outcome is desired, meaning passing a test or not having a disease.

To make our predicted classifications more useful, we then further disaggregate them into three groups: predicted to fail (PANCE <350), at-risk of failing (350-375), and likely to pass (>375). This is analogous to the “traffic signal” strategy (Arnold and Pistilli 2012). We argue that sorting students into these three groups is most useful for our goals of optimizing potential remediation well in advance of the exam. We present 3-by-3

confusion matrices that show all three classifications and discuss its practical application in our educational context.

Results

Table 2: Predictive model results based on 2-by-2 confusion matrix

Model	TP	FP	TN	FN	Accuracy	Sensitivity	Specificity
Adaptive Minimum Match KNN	9	9	158	4	0.93	0.69	0.95
Standard KNN*	9	17	150	4	0.88	0.69	0.90
Random Forest*	10	25	142	3	0.84	0.77	0.85
Support Vector Machine*	8	41	126	5	0.74	0.62	0.75

Best model with each predictive approach is displayed. Example interpretation: Adaptive Minimum Match KNN (AMMKNN) predicted 9 students as true positives, 9 false positives, 158 true negatives, and 4 false negatives. *The non-AMMKNN models required manual tuning of predicted results (after the model has been run) to be meaningful; this is further discussed in file *S1 code and data appendix*.

Table 2 shows the LOOCV results of multiple predictive models, based on standard 2-by-2 confusion matrices. AMMKNN has the highest accuracy of 0.93. Since our priority as educators is to minimize false negatives (students who “fall through the cracks”), we want to maximize sensitivity while keeping false positives within reasonable limits. The model with the highest sensitivity is Random Forest, which predicts that 35 students will fail, 10 of which are correct predictions and 25 of which are incorrect. Providing extra support to 35 students when only 10 of them (less than one-third) need it is not reasonable and would use too many resources. The next highest sensitivity is 0.69, shared by AMMKNN and standard KNN. AMMKNN also has higher accuracy and specificity. AMMKNN predicts that 18 students will fail, with 9 of these predictions being correct and 9 being incorrect. In this scenario, half of all students flagged for extra support would truly require it.

To more clearly break down the predictions, Table 3 shows 3-by-3 confusion matrices for our preferred model, AMMKNN, as well as the next best model, standard KNN. The accuracy for AMMKNN is calculated as the number of correct predictions divided by the number of total students: (9 + 2 + 123)/180 = 0.74. Even though this accuracy is lower

than the accuracy of the same model when a 2-by-2 matrix is used (0.93), we argue that the 3-by-3 version—which views results as a spectrum of classification categories—is more useful when considering the use of the model in practice.

We can build a framework based on the 3-by-3 AMMKNN results: 18 students are predicted to fail the PANCE. Out of these 18 predictions, 9 would be completely correct, 1 would be justifiable (because it would be acceptable to remediate a student who would otherwise almost fail), and 8 would be "unnecessary support." The model would identify 26 students as "at risk." Out of these 26 predictions, 4 would be useful (2 students who truly fail and 2 students "at-risk") and 22 would be "unnecessary support." The model would identify 136 students as likely to pass. Out of these 136 predictions, 123 would be "predictable no support," 11 would pass with an "at-risk" score, and 2 would "fall through the cracks." When we create the same framework for the standard KNN model, the results are less favorable (there are more "unnecessary support" students). Therefore, AMMKNN is the most useful model for our purposes, and might be useful to other educators with similar priorities.

Table 3: 3-by-3 confusion matrices for best predictive models

		Predicted scores		
		<350	350-375	>375
Actual scores	<350	9	2	2
	350-375	1	2	11
	>375	8	22	123

3a: Adaptive Minimum Match KNN results

		Predicted scores		
		<350	350-375	>375
Actual scores	<350	9	2	2
	350-375	1	1	12
	>375	16	7	130

3b: Standard KNN results

After completing the validation and inspection of results above, the final step is to put our best predictive model (AMMKNN) into action with currently-enrolled students. These students have not yet taken the PANCE and were not part of the development and testing of AMMKNN. Our model predicts that 3 currently-enrolled students will fail, 9 will be "at risk" and pass, and 32 are likely to pass. These predictions will inform remediation efforts.

Discussion

Our results demonstrate the value of the 3-by-3 approach from a practical standpoint, rather than a 2-by-2 approach which is focused strictly on accuracy. The 3-by-3 approach provided additional information that can be used in conjunction with other programmatic information to help students achieve success. We further argue that the use of this and similar predictive models needs to be considered from three perspectives—educator, student, and program administrator—which together lead to a practical and ethically-sound student support strategy. Balancing these perspectives involves tradeoffs that remind us that analytics are best used as one among multiple inputs to decision-making.

Educator perspective

Our top priority when building a predictive model was to reduce the number of FN, meaning students who "fall through the cracks." Accomplishing this comes at the cost of having many "unnecessary support" students also identified by the model. Therefore, it is critical to proceed cautiously when applying results from this or any other predictive model to our educational practice. When we apply this analytics process to predict student PANCE outcomes, the predictive model will label our students as likely to fail, be at-risk, or pass. Even though similar educational analytics success stories abound (Arnold and Pistilli 2012; Ekowo and Palmer 2016), there is also concerning evidence that labeling students as "failers" can be problematic (Barouch-Gilbert 2015; Kumar 2019). Framing of the results when communicating with the students is something that the educator must consider, so that the students understand the basis and uncertainties associated with actions suggested.

Given these hazards, we propose a student support strategy that uses predictive results in triangulation with other possible indicators of future student outcomes. We plan to compare the model's predictions to students that a) have already been flagged by advisors or instructors, b) belong to historically marginalized and underrepresented groups, and/or c) self-identify as requiring extra support. When we apply this method in practice, we plan to individually review all information that we have on the students who are predicted to fail or be at-risk. Then, taking all information into account, we plan to provide remedial support to a subset of these students, as they prepare for the PANCE exam.

Student perspective

This study does not include an empirical examination of student reactions to receiving analytics-based results. Therefore, we can only speculate about possible reactions and considerations from the student perspective. We believe that this is important to address, even though it may be incomplete.

We hope to incorporate considerations of student mental health and long-term outcomes into our proposed student support strategy, with a focus on transparency. When students are told that they have been identified for receiving remedial support, we anticipate that this could cause distress. We might be able to mitigate some of this distress by telling each student that a) other factors—such as course grades, concern from instructors, or known extenuating circumstances—also factored into our decision to offer remedial support, and b) the predictive model can make mistakes and identify "unnecessary support" (FP) students accidentally, the same way that a disease test can give a FP result to somebody who does not have the disease. We can also include student feedback in the process.

Additionally, student experiences of framing effects (Tversky and Kahneman 1981; McNeil et al. 1982) needs to be investigated and optimized. The term "remedial" might sound harsh. We might instead choose to use a term like "extra support" or "supplemental curriculum." Further work is required to solicit and incorporate student input into the use of analytics.

Program administrator perspective

From the perspective of a program administrator, leveraging this predictive model to make a successful student support strategy can be logistically and resource intensive. For our latest cohort, our predictive model results identify 3 students as likely to fail and 9 students as "at-risk" out of a cohort of 44 students. This means that 27 percent of students in this cohort might require extra instructor resources (which in turn might require additional funding). Time will also need to be carved out within these students' program of study to allow for them to engage in remedial activities. These considerations motivated us to only use first-year data when building a predictive model, so that students still have their second year in the program to prepare for the PANCE exam. This one year period would likely give program administrators enough time to arrange for the appropriate support for all students to the extent that resources allow. Additional analytics could also be conducted with second-year assessment data as students move closer to the exam.

We must use these predictions carefully, knowing that some of those predicted to fail could be "unnecessary support" (FP) students, but we of course do not know which ones. We will not know these students' true PANCE scores for another year.

Study limitations and strengths

While our analytics approach is useful as an in-house tool and might be applicable to broader educational settings, we acknowledge a number of limitations in our study and results. The ability of our AMMKNN model to generalize to other types of health professions programs is unknown. Therefore, we recommend that others using this approach conduct thorough validation after training the model and work to transparently apply the results when used for making predictions. Given our small sample size from four cohorts of students in a single program, we also cannot yet comment on whether our model's performance will improve or not after additional data are added.

The predictive model depends on collecting the same independent variables on students every year and other across-cohort uniformity. If curriculum changes or instructional practices lead to non-comparable data being collected across cohorts, our predictive approach might not work as effectively. Furthermore, the cohorts included in our study involved some students that were and others who were not affected by the COVID-19 pandemic. Some students did the PANCE exam in the middle of the pandemic, while most did it prior to the pandemic. We do not currently measure and include variables to account for these potential cohort-specific differences.

Our study also has some strengths to build upon in future work. Our ability to annually update the training data might address some cross-cohort variation. When we included a variable that identified specific cohorts, predictions did not improve, suggesting that some variation might be accommodated by our approach. Additionally, educational analytics are often plagued by small sample sizes. We developed the AMMKNN method to specifically address this problem and were able to make improvements over standard models, without the need for extensive fine tuning of model parameters. The method now needs to be tested with other data and contexts.

Conclusion

As educators, we hope to create a safety net around our students that can support them as needed, especially in health professions education with a) patient outcomes at stake and b)

delays in certification leading to unwanted professional consequences. The “adaptive minimum match” KNN model we have developed and validated serves as one additional tool that—along with already-existing tools—makes this safety net stronger when used responsibly and transparently. We conclude that a) educational analytics should prioritize identification of “likely-to-fail” or “at-risk” students over traditional metrics like accuracy, b) these analytics should co-exist alongside and support other approaches to student support, and c) educator, student, and program administrator perspectives all need to be taken into account for the practical and ethical implementation of educational analytics.

Acknowledgements

We would like to thank Taylor DiJohnson for preparing and managing the data for this project, coordinating between team members, and keeping everything organized. This project would not be possible without your tireless work and enthusiasm, Taylor! We would also like to thank Valay Maskey for assistance with preparing and formatting this article as well as putting the AdaptiveLearnalytics package online.

References

Akçapınar G, Altun A, Aşkar P. Using learning analytics to develop early-warning system for at-risk students. International Journal of Educational Technology in Higher Education. 2019; 16(1): 1-20. doi: 10.1186/s41239-019-0172-z.

Arnold KE, Pistilli MD. Course signals at Purdue: Using learning analytics to increase student success. Proceedings of the 2nd international conference on learning analytics and knowledge. 2020; 267-270. doi: 10.1145/2330601.2330666.

Aulck L, Velagapudi N, Blumenstock J, West J. Predicting student dropout in higher education. Presented at 2016 ICML Workshop on #Data4Good: Machine Learning in Social Good Applications, New York, NY. 2016. arXiv preprint arXiv:1606.06364.

Bissonnette V, Mirchi N, Ledwos N, Alsidieri G, Winkler-Schwartz A, Del Maestro R. F. Artificial intelligence distinguishes surgical training levels in a virtual reality spinal task. The Journal of bone and joint surgery. 2019; 101(23): e127. doi: 10.2106/JBJS.18.01197.

Black EW, Buchs SR, Garbas B. Using Data Mining for the Early Identification of Struggling Learners in Physician Assistant Education. The Journal of Physician Assistant Education. 2021; 32(1): 38-42. doi: 10.1097/JPA.0000000000000347.

Chan KS, Zary N. Applications and challenges of implementing artificial intelligence in medical education: integrative review. JMIR medical education. 2019; 5(1): e13930. doi: 10.2196/13930.

Chan AK, Botelho MG, Lam OL. The relation of online learning analytics, approaches to learning and academic achievement in a clinical skills course. European Journal of Dental Education. 2020; 25(3): 442-450. doi: 10.1111/eje.12619.

Ekowo M, Palmer I. The Promise and Peril of Predictive Analytics in Higher Education: A Landscape Analysis. New America. 2016. https://eric.ed.gov/?id=ED570869.

Gierl MJ, Latifi S, Lai H, Boulais AP, De Champlain A. Automated essay scoring and the future of educational assessment in medical education. Medical education. 2014; 48(10): 950-962. doi: 10.1111/medu.12517.

Kotsiantis SB. Use of machine learning techniques for educational proposes: a decision support system for forecasting students' grades. Artificial Intelligence Review. 2012; 37(4): 331-344. https://link.springer.com/article/10.1007/s10462-011-9234-x.

Kučak D, Juričić V, Đambić G. Machine Learning in Education - a Survey Of Current Research Trends. Annals of DAAAM & Proceedings, 29. 2018. doi: 10.2507/29th.daaam.proceedings.059.

Liz-Domínguez M, Rodríguez MC, Nistal ML, Mikic-Fonte FA. Predictors and early warning systems in higher education-A systematic literature review. LASI-SPAIN. 2019; 84-99. http://ceur-ws.org/Vol-2415/paper08.pdf.

Kibanov M, Becker M, Mueller J, Atzmueller M.,Hotho A, Stumme G. Adaptive kNN using expected accuracy for classification of geo-spatial data. In Proceedings of the 33rd Annual ACM Symposium on Applied Computing. 2018; 857-865. doi: 10.1145/3167132.3167226.

Kumar A. Cultures of learning in developing education systems: Government and NGO classrooms in India. International Journal of Educational Research. 2019; 95: 76-89. doi: 10.1016/j.ijer.2019.02.009.

Kumar A. Edwards RA. AdaptiveLearnalytics: Adaptive Predictive Learning Analytic Tools. 2021. https://github.com/readcreate/AdaptiveLearnalytics.

McMahon KK. Team-Based Learning. Chapter 5 in: Jeffries WB, Huggett K, editors. An introduction to medical teaching. Springer Science & Business Media; 2010.

McNeil BJ, Pauker SG, Sox Jr HC, Tversky A. On the elicitation of preferences for alternative therapies. New England journal of medicine. 1982; 306(21): 1259-1262. doi: 10.1056/NEJM198205273062103.

Michaelsen LK, Sweet M. The essential elements of team-based learning. New directions for teaching and learning. 2008; 116: 7-27. doi: 10.1002/tl.330.

Ougiaroglou S, Nanopoulos A, Papadopoulos AN, Manolopoulos Y, Welzer-Druzovec T. Adaptive k-nearest-neighbor classification using a dynamic number of nearest neighbors. East European Conference on Advances in Databases and Information Systems. 2007: 66-82. doi: 10.1007/978-3-540-75185-4_7.

R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. 2020. https://www.R-project.org/.

Rao RB, Fung G, Rosales R. On the dangers of cross-validation. An experimental evaluation. Proceedings of the 2008 SIAM international conference on data mining. 2008; 588-596. Society for Industrial and Applied Mathematics. doi: 10.1137/1.9781611972788.54.

Saqr M, Fors U, Tedre M. How learning analytics can early predict under-achieving students in a blended medical education course. Medical teacher. 2017; 39(7): 757-767. doi: 10.1080/0142159X.2017.1309376.

Sokkhey P, Okazaki T. Comparative Study of Prediction Models on High School Student Performance in Mathematics. 34th International Technical Conference on

Circuits/Systems, Computers and Communications. IEEE. 2019: 1-4. IEEE. doi: 10.1109/ITC-CSCC.2019.8793331.

Stimpson AJ, Cummings ML. Assessing intervention timing in computer-based education using machine learning algorithms. IEEE Access. 2014; 2: 78-87. doi: 10.1109/ACCESS.2014.2303071.

Sun S, Huang R. An adaptive k-nearest neighbor algorithm. Seventh international conference on fuzzy systems and knowledge discovery. IEEE. 2010; 1: 91-94. doi: 10.1109/FSKD.2010.5569740.

ten Cate O, Dahdal S, Lambert T, Neubauer F, Pless A, Pohlmann PF, et al. Ten caveats of learning analytics in health professions education: A consumer's perspective. Medical teacher. 2020; 42(6): 673-678. doi: 10.1080/0142159X.2020.1733505.

Tolsgaard MG, Boscardin CK, Park YS, Cuddy MM, Sebok-Syer SS. The role of data science and machine learning in Health Professions Education: practical applications, theoretical contributions, and epistemic beliefs. Advances in Health Sciences Education. 2020; 1-30. doi: 10.1007/s10459-020-10009-8.

Tversky A, Kahneman D. The framing of decisions and the psychology of choice. Science. 1981; 211(4481): 453-458. doi: 10.1126/science.7455683.

Waheed H, Hassan SU, Aljohani NR, Hardman J, Alelyani S, Nawaz R. Predicting academic performance of students from VLE big data using deep learning models. Computers in Human behavior. 2020; 104(106189). doi: 10.1016/j.chb.2019.106189.

Wallace V, Walker L. Team-based learning. Chapter 7 in: Kayingo G, Hass VM, editors. The health professions educator: A practical guide for new and established faculty. Springer Publishing Company; 2018.

Walkowski S, Lundin M, Szymas J, Lundin J. Exploring viewing behavior data from whole slide images to predict correctness of students' answers during practical exams in oral pathology. Journal of pathology informatics. 2015; 6. doi: 10.4103/2153-3539.158057.

Winkler-Schwartz A, Yilmaz R, Mirchi N, Bissonnette V, Ledwos N, Siyar S, et al. Machine learning identification of surgical and operative factors associated with surgical expertise in virtual reality simulation. JAMA network open. 2019; 2(8): e198363-e198363. doi: 10.1001/jamanetworkopen.2019.8363.

Zohair LMA. Prediction of Student's performance by modelling small dataset size. International Journal of Educational Technology in Higher Education. 2019; 16(1): 1-18. doi: 10.1186/s41239-019-0160-3.

Supporting information captions

S1 code and data appendix. *A PDF file containing all R code (which should be sufficient for replication), AdaptiveLearnalytics package information, full examples of predictive modeling process, and technical discussion about methods and results.*

S1 code and data appendix

Anshul Kumar, Lisa Walker, Roger Edwards

Version – 04 August 2021

Contents

1 Information and initial set-up

This supplementary code and data appendix file accompanies the article **The application of predictive analytics to identify at-risk students in health professions education** in 2021.

- Corresponding author: Anshul Kumar, akumar@mghihp.edu

- The R code for many of the functions/processes used in this study are stored in the R package that we have created for this analysis, called `AdaptiveLearnalytics`. Below, we install and load this package.

```r
if (!require(devtools)) install.packages('devtools')
library(devtools)
devtools::install_github("readcreate/AdaptiveLearnalytics")
```

```r
library(AdaptiveLearnalytics)
```

```
##
## Thanks for using the AdaptiveLearnalytics package.
##
## Please cite this package when you use it.
##
## We welcome your questions or feedback: Anshul Kumar <akumar@mghihp.edu>
##
## Suggested citation:
## Anshul Kumar and Roger Edwards. 2021. AdaptiveLearnalytics: Adaptive Predictive Learning Analytic To
##
## Run the following code for more information:
## help(package = 'AdaptiveLearnalytics')
##
## This package was initially developed to assist with the following publication: Anshul Kumar, Lisa Wa
##
## We used the following resources to make this package:
## * Erik Erhardt. 2018. R Package Development. https://statacumen.com/teach/ShortCourses/R_Packages/R_
## * Hadley Wickham. R Packages. https://r-pkgs.org/
## * Sharon Machlis. 2019. How to write your own R package. IDG TECHtalk. https://www.youtube.com/watch
```

Readers can use the code above to install and use the package.

The following code can be run to see more information:

```r
help(package = 'AdaptiveLearnalytics')
```

- Below, we set our working directory:

```r
setwd("path/to/working/directory")
```

(The code above is for illustrative purposes. We set our real working directory in a hidden code chunk).

- We are unfortunately not able to make our data publicly available. However, the code below combined with the examples within the `AdaptiveLearnalytics` package should be sufficient to reproduce our work methods on different data sets.

2 Load and prepare data

2.1 Load from Excel file

The code below subsets our data and produces output to help us confirm that the subsetting worked.

```
yearcutoff <- 2019 # writing 2018 here will include first 3 cohorts

# load data into R
library(readxl)
d <- read_excel("AI Research Template 20210308.xlsx")

dim(d)
```

```
## [1] 229 325
```

```
d <- d[which(d$`Year student began program`<yearcutoff),]

dim(d)
```

```
## [1] 183 325
```

```
d <- d[which(!is.na(d$PANCE)),]

dim(d)
```

```
## [1] 180 325
```

Now we have our initial data, as a starting point, called `d`.

We make a copy for later use:

```
CopyOfInitialData <- d
```

2.2 Manually eliminate unwanted variables

Below, we remove a few variables that we know we definitely don't want to include (mostly due to data being unavailable or specific variable coding).

```
# Variables were manually selected

d <- d[c(
  # "cohort",
  "Overall GPA"
  ,"Overall Science GPA"
  # ,"GRE Official Overall Score"
  # ,"GRE Analytical Scaled"
  # ,"GRE Quantitative Scaled"
  # ,"GRE Verbal Scaled"
  ,"Foundations.iRAT.mean"
  ,"Foundations CAT"
  ,"PA650 Foundations  of Med Final Grade"
  ,"PA Professions Exam 1"
  ,"PA Professions Exam 2"
  ,"Hematolog.iRAT.mean"
  ,"Hematology Online Anatomy Practical"
  ,"Hematology CAT"
```

```
,"PA732 Hematology Final Grade"
,"Derm.iRAT.mean"
,"Derm CAT"
,"PA730 Derm Final Grade"
,"MSK.iRAT.mean"
,"Practical 1"
,"MSK CAT"
,"PA731 Muscskl Ds & Injury Final Grade"
,"Neuro.iRAT.mean"
,"Neuro Anatomy Practical"
,"Neuro CAT"
,"PA734  Neurology Final Grade"
,"Patient Care Final Written Exam (CAT 1)"
,"Patient Care I Final Grade"
# ,"PA in Practice coding and billing quiz" # problematic
,"PA in Practice Written Exam 1"
,"PA in Practice  Practical Exam 2"
,"PA in Practice Written Exam 2"
# ,"PA in Practice Exam 4 Practical" # problematic
,"PA721  PA in Practice  Final Grade"
,"Cardio.iRAT.mean"
,"Cardio CAT #1"
,"PA736 Cardiovascular Disease Final Grade"
,"Pulm.iRAT.means"
,"Pulm CAT"
,"PA733 Pulmonary Med Final Grade"
,"Oto.iRAT.mean"
,"Otolar & OphthalAnatomy Practical"
,"Otolar & OphthalCAT"
,"PA737 Otolar & Ophthal Final Grade"
,"Behav.iRAT.mean"
,"Behav Med CAT"
,"PA735 Prin of Beh  Med Final Grade"
,"Patient Care II Exam 1"
,"PA751 Patient Care II Final Grade"
,"NephUro.iRAT.mean"
,"Neph/Uro CAT"
,"Endo Anatomy Practical"
,"Endo CAT"
,"PA739  Endocrinology Final Grade"
,"GI.iRAT.mean"
,"GI CAT"
,"GI Anatomy Practical"
,"PA738 Gastroenterlogy Final Grade"
,"Repro.iRAT.mean"
,"Repro Anatomy Practical"
,"Repro CAT"
,"PA741 Prin  of Repro Med Final Grade"
,"Pt Care III Exam 1"
,"PA752 Patient Care III Final Grade"
,"Special Pops Midterm"
,"Special Pops Final"
,"PA760 Special Pops Final Grade"
```

```
  ,"SEI.iRAT.mean"
  ,"SEI Written Exam"
  ,"PA770 Surg, EM, and IP Care Final Grade"
  ,"PACKRAT I  Raw Score"
  # ,"PACKRAT I Cardio"
  # ,"PACKRAT I Derm"
  # ,"PACKRAT I Endo"
  # ,"PACKRAT I ENT/Opht"
  # ,"PACKRAT I GI"
  # ,"PACKRAT I HEMATOLOGY"
  # ,"PACKRAT I INFECTIOUS DISEASES"
  # ,"PACKRAT I NEURO"
  # ,"PACKRAT I OBGYN"
  # ,"PACKRAT I ORTHO/RHEM"
  # ,"PACKRAT I PSYCH/BEHVMED"
  # ,"PACKRAT I PULM"
  # ,"PACKRAT I URO/REAL"
  # ,"PACKRAT I CLINICAL INTERVENTION"
  # ,"PACKRAT I CLINICAL THERAPEUTICS"
  # ,"PACKRAT I DIAGNOSIS"
  # ,"PACKRAT I DIAGNOSTIC STUDIES"
  # ,"PACKRAT I HEALTH MAINTENANCE"
  # ,"PACKRAT 1 HISTORY & PHYSICAL"
  # ,"PACKRAT I SCIENTIFIC CONCEPTS"
  ,"PANCE"
)]
```

2.3 Fix variable names

```
names(d) <- make.names(names(d))
```

2.4 Standardize data

```
library(jtools)
df<-jtools::standardize(d)

# put unstandardized dependent variable back into data
df$PANCE <- CopyOfInitialData$PANCE
```

2.5 Correlation to select independent variables

```
CorrelationThreshold <- .20
```

We want to remove variables from the data set that are correlated at lower than `CorrelationThreshold` with the dependent variable `PANCE`.

```r
df <- corVarSelect("PANCE",df,CorrelationThreshold)
```

```
##
##
## Starting process. Input dataframe is called df and contains 180 observations and 66 variables, inclu
##
##
##
## 1. Correlation between Overall.GPA (SD = 1) and PANCE = 0.2424909.
##
## 2. Correlation between Overall.Science.GPA (SD = 1) and PANCE = 0.2707181.
##
## 3. Correlation between Foundations.iRAT.mean (SD = 1) and PANCE = 0.4719581.
##
## 4. Correlation between Foundations.CAT (SD = 1) and PANCE = 0.353719.
##
## 5. Correlation between PA650.Foundations..of.Med.Final.Grade (SD = 1) and PANCE = 0.4063876.
##
## 6. Correlation between PA.Professions.Exam.1 (SD = 1) and PANCE = 0.1197356.
##
## 7. Correlation between PA.Professions.Exam.2 (SD = 1) and PANCE = -0.02407695.
##
## 8. Correlation between Hematolog.iRAT.mean (SD = 1) and PANCE = 0.488116.
##
## 9. Correlation between Hematology.Online.Anatomy.Practical (SD = 1) and PANCE = -0.1346972.
##
## 10. Correlation between Hematology.CAT (SD = 1) and PANCE = 0.4423458.
##
## 11. Correlation between PA732.Hematology.Final.Grade (SD = 1) and PANCE = 0.4907659.
##
## 12. Correlation between Derm.iRAT.mean (SD = 1) and PANCE = 0.3324565.
##
## 13. Correlation between Derm.CAT (SD = 1) and PANCE = 0.3308209.
##
## 14. Correlation between PA730.Derm.Final.Grade (SD = 1) and PANCE = 0.3116946.
##
## 15. Correlation between MSK.iRAT.mean (SD = 1) and PANCE = 0.3546153.
##
## 16. Correlation between Practical.1 (SD = 1) and PANCE = 0.2951705.
##
## 17. Correlation between MSK.CAT (SD = 1) and PANCE = 0.3208405.
##
## 18. Correlation between PA731.Muscskl.Ds...Injury.Final.Grade (SD = 1) and PANCE = 0.3720635.
##
## 19. Correlation between Neuro.iRAT.mean (SD = 1) and PANCE = 0.4793173.
##
## 20. Correlation between Neuro.Anatomy.Practical (SD = 1) and PANCE = 0.2032043.
##
## 21. Correlation between Neuro.CAT (SD = 1) and PANCE = 0.3787337.
##
## 22. Correlation between PA734..Neurology.Final.Grade (SD = 1) and PANCE = 0.4139737.
##
## 23. Correlation between Patient.Care.Final.Written.Exam..CAT.1. (SD = 1) and PANCE = 0.4215722.
```

```
##
## 24. Correlation between Patient.Care.I.Final.Grade (SD = 1) and PANCE = 0.266896.
##
## 25. Correlation between PA.in.Practice.Written.Exam.1 (SD = 1) and PANCE = 0.07550383.
##
## 26. Correlation between PA.in.Practice..Practical.Exam.2 (SD = 1) and PANCE = 0.0009872979.
##
## 27. Correlation between PA.in.Practice.Written.Exam.2 (SD = 1) and PANCE = 0.01953831.
##
## 28. Correlation between PA721..PA.in.Practice..Final.Grade (SD = 1) and PANCE = 0.03055977.
##
## 29. Correlation between Cardio.iRAT.mean (SD = 1) and PANCE = 0.5139086.
##
## 30. Correlation between Cardio.CAT..1 (SD = 1) and PANCE = 0.4980982.
##
## 31. Correlation between PA736.Cardiovascular.Disease.Final.Grade (SD = 1) and PANCE = 0.5317535.
##
## 32. Correlation between Pulm.iRAT.means (SD = 1) and PANCE = 0.4630837.
##
## 33. Correlation between Pulm.CAT (SD = 1) and PANCE = 0.4983239.
##
## 34. Correlation between PA733.Pulmonary.Med.Final.Grade (SD = 1) and PANCE = 0.5425222.
##
## 35. Correlation between Oto.iRAT.mean (SD = 1) and PANCE = 0.4343808.
##
## 36. Correlation between Otolar...OphthalAnatomy.Practical (SD = 1) and PANCE = 0.1526747.
##
## 37. Correlation between Otolar...OphthalCAT (SD = 1) and PANCE = 0.2008488.
##
## 38. Correlation between PA737.Otolar...Ophthal.Final.Grade (SD = 1) and PANCE = 0.3087027.
##
## 39. Correlation between Behav.iRAT.mean (SD = 1) and PANCE = 0.3860884.
##
## 40. Correlation between Behav.Med.CAT (SD = 1) and PANCE = 0.2930686.
##
## 41. Correlation between PA735.Prin.of.Beh..Med.Final.Grade (SD = 1) and PANCE = 0.352443.
##
## 42. Correlation between Patient.Care.II.Exam.1 (SD = 1) and PANCE = 0.255107.
##
## 43. Correlation between PA751.Patient.Care.II.Final.Grade (SD = 1) and PANCE = 0.2657109.
##
## 44. Correlation between NephUro.iRAT.mean (SD = 1) and PANCE = 0.272454.
##
## 45. Correlation between Neph.Uro.CAT (SD = 1) and PANCE = 0.2703458.
##
## 46. Correlation between Endo.Anatomy.Practical (SD = 1) and PANCE = 0.1712717.
##
## 47. Correlation between Endo.CAT (SD = 1) and PANCE = 0.3678981.
##
## 48. Correlation between PA739..Endocrinology.Final.Grade (SD = 1) and PANCE = 0.4192088.
##
## 49. Correlation between GI.iRAT.mean (SD = 1) and PANCE = 0.2162198.
##
## 50. Correlation between GI.CAT (SD = 1) and PANCE = 0.3549343.
```

```
##
## 51. Correlation between GI.Anatomy.Practical (SD = 1) and PANCE = 0.07031748.
##
## 52. Correlation between PA738.Gastroenterlogy.Final.Grade (SD = 1) and PANCE = 0.4079875.
##
## 53. Correlation between Repro.iRAT.mean (SD = 1) and PANCE = 0.4343981.
##
## 54. Correlation between Repro.Anatomy.Practical (SD = 1) and PANCE = 0.1451081.
##
## 55. Correlation between Repro.CAT (SD = 1) and PANCE = 0.4616046.
##
## 56. Correlation between PA741.Prin..of.Repro.Med.Final.Grade (SD = 1) and PANCE = 0.3932625.
##
## 57. Correlation between Pt.Care.III.Exam.1 (SD = 1) and PANCE = 0.2125311.
##
## 58. Correlation between PA752.Patient.Care.III.Final.Grade (SD = 1) and PANCE = 0.2647794.
##
## 59. Correlation between Special.Pops.Midterm (SD = 1) and PANCE = 0.2939934.
##
## 60. Correlation between Special.Pops.Final (SD = 1) and PANCE = 0.2679239.
##
## 61. Correlation between PA760.Special.Pops.Final.Grade (SD = 1) and PANCE = 0.3627844.
##
## 62. Correlation between SEI.iRAT.mean (SD = 1) and PANCE = 0.4244655.
##
## 63. Correlation between SEI.Written.Exam (SD = 1) and PANCE = 0.5101119.
##
## 64. Correlation between PA770.Surg..EM..and.IP.Care.Final.Grade (SD = 1) and PANCE = 0.5654293.
##
## 65. Correlation between PACKRAT.I..Raw.Score (SD = 1) and PANCE = 0.6466527.
##
## 66. Correlation between PANCE (SD = 64.01432) and PANCE = 1.
##
## Variable selection process is complete. Returned dataframe contains 180 observations and 55 variables
```

Make a copy of the new data set, with correlation-selected variables:

```
CopyOfDataAfterCor <- df
```

2.6 Final list of variables

```
names(df)
```

```
##  [1] "Overall.GPA"
##  [2] "Overall.Science.GPA"
##  [3] "Foundations.iRAT.mean"
##  [4] "Foundations.CAT"
##  [5] "PA650.Foundations..of.Med.Final.Grade"
##  [6] "Hematolog.iRAT.mean"
##  [7] "Hematology.CAT"
##  [8] "PA732.Hematology.Final.Grade"
```

```
##  [9] "Derm.iRAT.mean"
## [10] "Derm.CAT"
## [11] "PA730.Derm.Final.Grade"
## [12] "MSK.iRAT.mean"
## [13] "Practical.1"
## [14] "MSK.CAT"
## [15] "PA731.Muscskl.Ds...Injury.Final.Grade"
## [16] "Neuro.iRAT.mean"
## [17] "Neuro.Anatomy.Practical"
## [18] "Neuro.CAT"
## [19] "PA734..Neurology.Final.Grade"
## [20] "Patient.Care.Final.Written.Exam..CAT.1."
## [21] "Patient.Care.I.Final.Grade"
## [22] "Cardio.iRAT.mean"
## [23] "Cardio.CAT..1"
## [24] "PA736.Cardiovascular.Disease.Final.Grade"
## [25] "Pulm.iRAT.means"
## [26] "Pulm.CAT"
## [27] "PA733.Pulmonary.Med.Final.Grade"
## [28] "Oto.iRAT.mean"
## [29] "Otolar...OphthalCAT"
## [30] "PA737.Otolar...Ophthal.Final.Grade"
## [31] "Behav.iRAT.mean"
## [32] "Behav.Med.CAT"
## [33] "PA735.Prin.of.Beh..Med.Final.Grade"
## [34] "Patient.Care.II.Exam.1"
## [35] "PA751.Patient.Care.II.Final.Grade"
## [36] "NephUro.iRAT.mean"
## [37] "Neph.Uro.CAT"
## [38] "Endo.CAT"
## [39] "PA739..Endocrinology.Final.Grade"
## [40] "GI.iRAT.mean"
## [41] "GI.CAT"
## [42] "PA738.Gastroenterlogy.Final.Grade"
## [43] "Repro.iRAT.mean"
## [44] "Repro.CAT"
## [45] "PA741.Prin..of.Repro.Med.Final.Grade"
## [46] "Pt.Care.III.Exam.1"
## [47] "PA752.Patient.Care.III.Final.Grade"
## [48] "Special.Pops.Midterm"
## [49] "Special.Pops.Final"
## [50] "PA760.Special.Pops.Final.Grade"
## [51] "SEI.iRAT.mean"
## [52] "SEI.Written.Exam"
## [53] "PA770.Surg..EM..and.IP.Care.Final.Grade"
## [54] "PACKRAT.I..Raw.Score"
## [55] "PANCE"
```

3 Build and test predictive models

3.1 Random forest

Train model with random train-test split:

```
trainingRowIndex <- sample(1:nrow(df), 0.75*nrow(df))
dtrain <- df[trainingRowIndex, ]  # model training data
dtest  <- df[-trainingRowIndex, ]   # model test data

library(randomForest)
rf1 <- randomForest(PANCE ~ .,data=dtrain, proximity=TRUE,ntree=1000)
```

Inspect results:

```
dtest$rf.pred <- predict(rf1, newdata = dtest)

dtest$PassPANCE <- ifelse(dtest$PANCE>349,1,0)

dtest$rf1.pred.PassPANCE <- ifelse(dtest$rf.pred > 390, 1, 0)

(cm1 <- with(dtest,table(PassPANCE,rf1.pred.PassPANCE)))
```

```
##          rf1.pred.PassPANCE
## PassPANCE  0  1
##         0  1  1
##         1  6 37
```

Train model with LOOCV:

```
df.rf.LOOCV <-
  crossValidate(df,
  "rf1 <- randomForest(PANCE ~ ., data=traindata, proximity=TRUE,ntree=1000);
  testdata$rf.pred <- predict(rf1, newdata = testdata)",
  nrow(df))
```

Inspect results:

```
df.rf.LOOCV$PassPANCE <- ifelse(df.rf.LOOCV$PANCE>349,1,0)

df.rf.LOOCV$rf1.pred.PassPANCE <-
  ifelse(df.rf.LOOCV$rf.pred > 349, 1, 0)

(cm1 <- with(df.rf.LOOCV,table(PassPANCE,rf1.pred.PassPANCE)))
```

```
##          rf1.pred.PassPANCE
## PassPANCE   1
##         0  13
##         1 167
```

```
df.rf.LOOCV$PassPANCE <- ifelse(df.rf.LOOCV$PANCE>349,1,0)

df.rf.LOOCV$rf1.pred.PassPANCE <- ifelse(df.rf.LOOCV$rf.pred > 390, 1, 0)

(cm1 <- with(df.rf.LOOCV,table(PassPANCE,rf1.pred.PassPANCE)))
```

```
##          rf1.pred.PassPANCE
## PassPANCE   0   1
##         0   6   7
##         1  10 157
```

```
df.rf.LOOCV$rf1.pred.PassPANCE <- ifelse(df.rf.LOOCV$rf.pred > 400, 1, 0)

(cm1 <- with(df.rf.LOOCV,table(PassPANCE,rf1.pred.PassPANCE)))
```

```
##          rf1.pred.PassPANCE
## PassPANCE   0   1
##         0  10   3
##         1  24 143
```

```
df.rf.LOOCV$rf1.pred.PassPANCE <- ifelse(df.rf.LOOCV$rf.pred > 410, 1, 0)

(cm1 <- with(df.rf.LOOCV,table(PassPANCE,rf1.pred.PassPANCE)))
```

```
##          rf1.pred.PassPANCE
## PassPANCE   0   1
##         0  10   3
##         1  49 118
```

3.2 Support vector machine

Train model with random train-test split:

```
trainingRowIndex <- sample(1:nrow(df), 0.75*nrow(df))
dtrain <- df[trainingRowIndex, ]  # model training data
dtest  <- df[-trainingRowIndex, ]   # model test data

library(e1071)
svm1 <-
  svm(PANCE ~ ., data = dtrain, kernel = "linear", cost = 10, scale = FALSE)
```

Inspect results:

```
dtest$svm1.pred <- predict(svm1, newdata = dtest)

dtest$PassPANCE <- ifelse(dtest$PANCE>349,1,0)

dtest$svm1.pred.PassPANCE <- ifelse(dtest$svm1.pred > 390, 1, 0)

(cm1 <- with(dtest,table(PassPANCE,svm1.pred.PassPANCE)))
```

```
##          svm1.pred.PassPANCE
## PassPANCE  0  1
##         0  1  4
##         1  7 33
```

Train model with LOOCV:

```
df.svm.LOOCV <-
  crossValidate(df,
                'svm1 <- svm(PANCE ~ ., data = traindata, kernel = "linear",
                cost = 10, scale = FALSE);
                testdata$svm1.pred <- predict(svm1, newdata = testdata)',
                nrow(df))
```

Inspect results:

```
df.svm.LOOCV$PassPANCE <- ifelse(df.svm.LOOCV$PANCE>349,1,0)

df.svm.LOOCV$svm1.pred.PassPANCE <- ifelse(df.svm.LOOCV$svm1.pred >349, 1, 0)

(cm1 <- with(df.svm.LOOCV,table(PassPANCE,svm1.pred.PassPANCE)))
```

```
##          svm1.pred.PassPANCE
## PassPANCE   0   1
##         0   4   9
##         1   6 161
```

```
df.svm.LOOCV$svm1.pred.PassPANCE <- ifelse(df.svm.LOOCV$svm1.pred > 390, 1, 0)

(cm1 <- with(df.svm.LOOCV,table(PassPANCE,svm1.pred.PassPANCE)))
```

```
##          svm1.pred.PassPANCE
## PassPANCE   0   1
##         0   6   7
##         1  29 138
```

```
df.svm.LOOCV$svm1.pred.PassPANCE <- ifelse(df.svm.LOOCV$svm1.pred > 400, 1, 0)

(cm1 <- with(df.svm.LOOCV,table(PassPANCE,svm1.pred.PassPANCE)))
```

```
##          svm1.pred.PassPANCE
## PassPANCE   0   1
##         0   8   5
##         1  41 126
```

```
df.svm.LOOCV$svm1.pred.PassPANCE <- ifelse(df.svm.LOOCV$svm1.pred > 410, 1, 0)

(cm1 <- with(df.svm.LOOCV,table(PassPANCE,svm1.pred.PassPANCE)))
```

```
##          svm1.pred.PassPANCE
## PassPANCE   0   1
##         0   8   5
##         1  49 118
```

3.3 Standard k-nearest neighbors

Train model with random train-test split:

```
trainingRowIndex <- sample(1:nrow(df), 0.75*nrow(df))
dtrain <- df[trainingRowIndex, ]  # model training data
dtest  <- df[-trainingRowIndex, ]   # model test data

library(dplyr)

dtrain.x <- dtrain %>% select(-PANCE) # remove DV
dtrain.y <- dtrain %>% select(PANCE) # keep only DV

dtest.x <- dtest %>% select(-PANCE) # remove DV
dtest.y <- dtest %>% select(PANCE) # keep only DV
```

```
sqrt(nrow(dtrain.x))
```

```
## [1] 11.61895
```

- We initially set `k` as 12, since it is the approximate square root of the number of observations in the training data.

```
library(FNN)
pred.a <- FNN::knn.reg(dtrain.x, dtest.x, dtrain.y, k = 12)
```

Inspect results:

```
dtest.y$knn12.pred <- pred.a$pred

dtest.y$PassPANCE <- ifelse(dtest.y$PANCE>349,1,0)

dtest.y$knn12.pred.PassPANCE <- ifelse(dtest.y$knn12.pred > 390, 1, 0)

(cm1 <- with(dtest.y,table(PassPANCE,knn12.pred.PassPANCE)))
```

```
##           knn12.pred.PassPANCE
## PassPANCE  0  1
##         0  1  3
##         1  1 40
```

Train model with LOOCV, k = 12:

```
library(dplyr)

df.knn12.LOOCV <-
  crossValidate(df, 'dtrain.x <- traindata %>% select(-PANCE);
dtrain.y <- traindata %>% select(PANCE);
dtest.x <- testdata %>% select(-PANCE);
dtest.y <- testdata %>% select(PANCE);
pred.a <- FNN::knn.reg(dtrain.x, dtest.x, dtrain.y, k = 12);
testdata$knn12.pred <- pred.a$pred', nrow(df))
```

Inspect results:

```
df.knn12.LOOCV$PassPANCE <- ifelse(df.knn12.LOOCV$PANCE>349,1,0)

df.knn12.LOOCV$knn12.pred.PassPANCE <-
  ifelse(df.knn12.LOOCV$knn12.pred >349, 1, 0)

(cm1 <- with(df.knn12.LOOCV,table(PassPANCE,knn12.pred.PassPANCE)))
```

```
##          knn12.pred.PassPANCE
## PassPANCE   1
##         0  13
##         1 167
```

```
df.knn12.LOOCV$knn12.pred.PassPANCE <-
  ifelse(df.knn12.LOOCV$knn12.pred > 390, 1, 0)

(cm1 <- with(df.knn12.LOOCV,table(PassPANCE,knn12.pred.PassPANCE)))
```

```
##          knn12.pred.PassPANCE
## PassPANCE   0   1
##         0   7   6
##         1   6 161
```

```
df.knn12.LOOCV$knn12.pred.PassPANCE <-
  ifelse(df.knn12.LOOCV$knn12.pred > 400, 1, 0)

(cm1 <- with(df.knn12.LOOCV,table(PassPANCE,knn12.pred.PassPANCE)))
```

```
##          knn12.pred.PassPANCE
## PassPANCE   0   1
##         0   9   4
##         1  17 150
```

Since the result above—with the prediction cut-off at 400—is decent, we disaggregate:

```
`Actual Values` <- cut(df.knn12.LOOCV$PANCE,
                       breaks=c(-Inf,350,375,Inf),
                       labels=c("<350","350-375",">375"))

`Predicted Values` <- cut(df.knn12.LOOCV$knn12.pred,
                          breaks=c(-Inf,400,405,Inf),
                          labels=c("<350","350-375",">375"))

# (cm<- addmargins(table(`Actual Values`,`Predicted Values`)))

(cm1<- table(`Actual Values`,`Predicted Values`))
```

```
##              Predicted Values
## Actual Values <350 350-375 >375
##       <350       9       2    2
##       350-375    1       1   12
##       >375      16       7  130
```

Above, when k=12, the predictions are not bad, but 16 students who truly score above 375 are predicted to fail, which is too many. Adaptive Minimum Match KNN (shown below) only predicts 8 students in this category. Furthermore, the standard KNN result above required careful adjustment —with prediction cut-offs at 400 and 405—before the results were useful. Adaptive Minimum Match KNN doesn't require such adjustment. These characteristics were also present in the disaggregated confusion matrix when k was set to 6 and 18.

```
df.knn12.LOOCV$knn12.pred.PassPANCE <-
  ifelse(df.knn12.LOOCV$knn12.pred > 410, 1, 0)

(cm1 <- with(df.knn12.LOOCV,table(PassPANCE,knn12.pred.PassPANCE)))
```

```
##          knn12.pred.PassPANCE
## PassPANCE   0   1
##         0  11   2
##         1  41 126
```

Although it is not shown, we have tried a range of values of `k` for the standard version of KNN and the results do not improve.

3.4 Adaptive minimum match k-nearest neighbors (AMMKNN)

Train model with random train-test split:

```
trainingRowIndex <- sample(1:nrow(df), 0.75*nrow(df))
dtrain <- df[trainingRowIndex, ]  # model training data
dtest  <- df[-trainingRowIndex, ]   # model test data

library(dplyr)

dtrain.x <- dtrain %>% select(-PANCE) # remove DV
dtrain.y <- dtrain %>% select(PANCE) # keep only DV

dtest.x <- dtest %>% select(-PANCE) # remove DV
dtest.y <- dtest %>% select(PANCE) # keep only DV
```

```
adaKNNminMatch1 <-
  adaptiveMinMatchKNNregression(dtrain.x, dtrain.y, dtest.x, maxK = 20)
```

Inspect results:

```
adaKNNminMatch1$PassPANCE <- ifelse(dtest.y$PANCE>349,1,0)

adaKNNminMatch1$MatchDVMeanMin.pred.PassPANCE <-
  ifelse(adaKNNminMatch1$MatchDVMeanMin >349, 1, 0)

(cm1 <- with(adaKNNminMatch1, table(PassPANCE,MatchDVMeanMin.pred.PassPANCE)))
```

```
##          MatchDVMeanMin.pred.PassPANCE
## PassPANCE  0  1
##         0  1  0
##         1  2 42
```

Train model with LOOCV:

```
library(dplyr)

adaKNNminMatch.LOOCV <-
  crossValidate(
    df,
    'dtrain.x <- traindata %>% select(-PANCE);
    dtrain.y <- traindata %>% select(PANCE);
    dtest.x <- testdata %>% select(-PANCE);
    pred.a <- adaptiveMinMatchKNNregression(
    dtrain.x, dtrain.y, dtest.x, maxK = 20);
    testdata <- pred.a', nrow(df))

# testdata$adaKNNminMatch.LOOCV.pred <- pred.a$MatchDVMeanMin
```

- In this and many other situations, the value of `maxK` might not matter too much. We suspect that this is because the mean of all matches will stabilize as the number of matches increases. In other words: the difference between the mean of the first 11 matches and the mean of the first 10 matches is likely smaller than the difference between the mean of the first 4 matches and the mean of the first 3 matches.

Inspect results:

```
# Add DV into predictive results
adaKNNminMatch.LOOCV$PANCE <- df$PANCE

# Make true pass/fail DV column
adaKNNminMatch.LOOCV$PassPANCE <- ifelse(adaKNNminMatch.LOOCV$PANCE>349, 1,0)

# Make predicted DV column
# those who scored less than 2 SDs below the mean on PACKRAT I
# are predicted to score the minimum of all matches (with no
# averaging involved);
# Everyone else is predicted to score the minimum of means;
adaKNNminMatch.LOOCV$pred.adj.PANCE <-
  ifelse(
    adaKNNminMatch.LOOCV$PACKRAT.I..Raw.Score < -2,
    adaKNNminMatch.LOOCV$MatchDVMin,
    adaKNNminMatch.LOOCV$MatchDVMeanMin)


# Make predicted pass/fail DV column
adaKNNminMatch.LOOCV$pred.adj.PassPANCE <-
  ifelse(adaKNNminMatch.LOOCV$pred.adj.PANCE > 349, 1, 0)

(cm1 <- with(adaKNNminMatch.LOOCV,table(PassPANCE,pred.adj.PassPANCE)))
```

```
##          pred.adj.PassPANCE
## PassPANCE   0   1
##         0   9   4
##         1   9 158
```

Changing the prediction threshold is not necessary, this time:

```
adaKNNminMatch.LOOCV$pred.adj.PassPANCE <-
  ifelse(adaKNNminMatch.LOOCV$pred.adj.PANCE > 370, 1, 0)

(cm1 <- with(adaKNNminMatch.LOOCV,table(PassPANCE,pred.adj.PassPANCE)))
```

```
##          pred.adj.PassPANCE
## PassPANCE   0   1
##         0  10   3
##         1  23 144
```

Disaggregated prediction:

```
`Actual Values` <-
  cut(
    adaKNNminMatch.LOOCV$PANCE,
    breaks=c(-Inf,350,375,Inf),
    labels=c("<350","350-375",">375"))

`Predicted Values` <-
  cut(adaKNNminMatch.LOOCV$pred.adj.PANCE,
      breaks=c(-Inf,350,375,Inf),
      labels=c("<350","350-375",">375"))

# (cm<- addmargins(table(`Actual Values`,`Predicted Values`)))

(cm1<- table(`Actual Values`,`Predicted Values`))
```

```
##              Predicted Values
## Actual Values <350 350-375 >375
##       <350       9       2    2
##       350-375    1       2   11
##       >375       8      22  123
```

4 Make new predictions

The best model, adaptive minimum match KNN, can now be used to make predictions on new students, as the final step of the entire analytics process.

Note that these new predictions were made using data from the latest cohort of students with *only the relevant independent variables that were available* at the time of publication. Once all relevant independent variables become available, we will of course re-run the procedure below with the complete data set.

Import and prepare new students' data:

```
library(readxl)
newStudents <-
  read_excel("AI Research Template 20210308 - Copy with 2019 packrat.xlsx")

newStudents <-
  newStudents[which(newStudents$`Year student began program`==2019),]
```

```
names(newStudents) <- make.names(names(newStudents))

newStudents <- newStudents[names(df)]

newStudents <- newStudents[ , colSums(is.na(newStudents)) < nrow(newStudents)]

newStudents <- na.omit(newStudents)

library(jtools)
newStudents<-jtools::standardize(newStudents)
```

Prepare training data:

```
dtrain.x <- df[names(newStudents)]

dtrain.y <- df[c("PANCE")]
```

Make predictions:

```
adaKNNminMatch.new <-
  adaptiveMinMatchKNNregression(dtrain.x, dtrain.y, newStudents, maxK = 20)
```

Process predictions:

```
adaKNNminMatch.new$pred.adj.PANCE <-
  ifelse(
    adaKNNminMatch.new$PACKRAT.I..Raw.Score < -2,
    adaKNNminMatch.new$MatchDVMin,
    adaKNNminMatch.new$MatchDVMeanMin)
```

```
# Make predicted pass/fail DV column
adaKNNminMatch.new$pred.adj.PassPANCE <-
  ifelse(adaKNNminMatch.new$pred.adj.PANCE > 349, 1, 0)
```

View results:

```
`Predicted Values` <-
  cut(adaKNNminMatch.new$pred.adj.PANCE,
      breaks=c(-Inf,350,375,Inf),
      labels=c("<350","350-375",">375"))

table(`Predicted Values`)
```

```
## Predicted Values
##    <350 350-375    >375
##       3       9      32
```

We, the users of these predictive tools/methods, can of course then sort the file `adaKNNminMatch.new` by the column `pred.adj.PANCE`. We will then match each row containing student data to student names. We will then have a list of the students are are at high risk, medium risk, and no risk of failing the certification exam. Generating this list and knowing its likely accuracy is the goal of our entire project.

5 Comparison of methods and models

Below, we offer some notes and commentary regarding the predictive methods we used and the results.

- For all of the standard models presented—and the many more that we attempted to use—a great deal of fine-tuning is required before results are even slightly useful. For example, and as shown above, when interpreting predictions from standard models, we find that the predictions need to be recalibrated. A predicted PANCE score of 400 from standard models is often the best cut-off threshold between true passing and failing, for the standard methods. With the AMMKNN process we developed, such recalibration does not (so far) appear to be needed.
- No classification techniques are shown in this document. Only regression techniques are shown. We did attempt a number of standard classification techniques and always found them to be deficient compared to regression techniques. This is not surprising to us, given that our dependent variable—even when considered as a discrete construct—is nevertheless a ranked variable with a meaningful order/hierarchy.
- The AMMKNN model that we develop is likely not feasible for use on large data sets. Since we have a small data set of 180 students, we can use it and it will not take too long to run. We have not yet done tests on massive data sets, but we suspect that the computing time and power needed would be prohibitive. More work is required on this issue.