Using Machine Learning to Identify At-risk Students in an Introductory Programming Course

Cameron I. Cooper (cooperc@sanjuancollege.edu)
San Juan College https://orcid.org/0000-0002-8062-5718
Kamea J. Cooper
San Juan College

Research Article

Keywords: computer science, early alert, early alert triggers, machine learning, student success, neural networks, gateway course

Posted Date: November 11th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1025335/v3

Abstract

Nationally, more than one-third of students enrolling in introductory computer science programming courses (CS101) do not succeed. To improve student success rates, this research team used supervised machine learning to identify students who are “at-risk” of not succeeding in CS101 at a two-year public college. The resultant predictive model accurately identifies $\approx$ 99% of “at-risk” students in an out-of-sample test data set. The programming instructor piloted the use of the model’s predictive factors as early alert triggers to intervene with individualized outreach and support across three course sections of CS101 in fall 2020. The outcome of this pilot study was a 23% increase in student success and a 7.3 percentage point decrease in DFW rate. More importantly, this study identified academic, early alert triggers for CS101. Specifically, the first two graded programs are of paramount importance for student success in the course.

Introduction

Bennedsen and Caspersen (2007) found student success rates at colleges and universities in CS101 to be approximately 67%. Seven years later, via meta-analysis, Watson and Li (2014) found the student success rates in CS101 to be essentially unchanged at 67%. This article discusses the development of an accurate early alert system using a neural network-based predictive system. Specifically, this system utilizes a probabilistic neural network to accurately identify students who are “at-risk” of not succeeding in their introduction to a programming course. The authors define at-risk outcomes as any course grade less than or equal to a 72% course average. The research found five graded measures (i.e., predictive factors), which combined to provide accurate predictions for students who were unlikely to succeed in CS101. These measures can then be treated as triggers for an early alert system that allows an instructor to approach an identified “at-risk” student with extra one-on-one course assistance with the goal of reroute the trajectory of the student toward course success.

The programming instructor piloted the early alert system during fall 2020. The pilot implementation of the early alert system described in this article resulted in a 7.3 percentage point decrease in D grade, Fail, Withdraw (DFW) rate and a 23% increase in student success for CS101 at the researchers’ home institution.

Background

According to the College’s Strategic Plan Annual: 2020-21, one of the strategic goals of the authors’ home institution is to “create an agile and responsive business model that responds to economic changes and focuses on helping all students achieve a high level of success in learning, completion.” This research directly facilitates the attainment of this goal by potentially helping computer science students be successful in the most significant gateway course in the two-year Associates Degree program at the school. The average student success rate in CS101 at college historically stands at 61.8%, five percentage points below the national average. With almost 40% of students willing to consider computer

science by taking CS101 unable to move onto the next course, the importance of improving the success rate in introductory computer science becomes more pressing, especially given the economic need for software developers. According to the U.S. Bureau of Labor Statistics, the job outlook is expected to grow by 22% over the next 10 years U.S. Bureau of Labor Statistics (2021). The opportunity cost for students unable to advance in a field ranked as the best job according to U.S. News and World Report is substantial (v). Efforts to improve student success must be undertaken. This research proposes an early alert system where as the academic semester progresses, key assignments trigger alerts for an instructor to step in and intervene. Ideally, interventions should occur early enough during the semester to help improve student outcomes by the end of the semester.

Literature Review

The use of neural networks as a means to predict student success has been employed in numerous contexts dating back to the mid-1990s. Hardgrave and Wilson (1994) used neural networks to predict graduate student success. Naik and Ragothaman (2004) utilized neural networks to predict MBA student success. More recently, the mentor for this project found neural networks to be an effective method in predicting student success in developmental mathematics and thereby improving student success at a four-year public institution of higher education in 2007. In 2008, van Heerden, Aldrich, and du Plessis (2008) demonstrated the ability of neural networks to predict student success in medical school.

Hanover Research offers a comprehensive overview of Early Alert Systems in Higher Education (2014). Important findings from Hanover relevant to this research include the following:

1. “Early alert systems may be most effective when targeting specific student populations, such as…at-risk students.” (p. 3)
2. An early alert system “entails a ‘systematic program’ that comprises at least ‘two key components’: alerts and intervention.” (p. 5)

This study focuses on the former. This research contributes to the body of student success literature by means of how ‘alerts’ (i.e., triggers) are determined. The authors feel the accuracy of the alert component of an effective system is vital to the system’s success. The authors employ neural networks as the means to accurately classify students as either at-risk or not at-risk. Thus, the most impactful factors/inputs into the neural network will then be treated as triggers for the early alert system.

1. Early alert systems are utilized by the majority of institutions of higher education (p. 6). Specifically, Noel Levitz found that 87.5% of public, two-year colleges have early alert systems in place. However, only 57.1% of these schools found their systems to be “very or somewhat effective.” This research aims to improve the efficacy of early alert systems at the course level and hopefully improve the 57.1% perceived efficacy at two-year institutions.
2. Metrics/factors to consider in predictive systems can be categorized as either “pre-enrollment factors” or “postenrollment” (p. 11). This research utilizes postenrollment factors (i.e., student ctory Computer Science).

PNNs, probabilistic neural networks, the type of neural network employed for this research, have been shown to be accurate in many diverse contexts, such as in stock market index forecasting (Kim & Hak Chun, 1998)), various signal processing applications (Zaknich, 1998), plant classification using leaf structures (Wu et al., 2007), and predicting bankruptcy (Yang et al., 1999). This research demonstrates the applicability of a PNN to accurately predict student success to assist targeting interventions.

An additional outcome from this research is the identification of factors that predict student success in introductory computer science courses. The identification of predictive factors impacting student success has been the topic of multiple researchers. Dalton, Moore, and Whittaker (2009) studied the impact of being a first generation and low income on student success. Karen Hamman (2016) published a study of factors that contribute to academic recovery. Millea, Willis, Elder, and Molina (2018) presented factors determining college retention and graduation rates.

After determining the predictive factors and constructing an accurate predictive system, an early alert system needs to be employed to improve student outcomes. Akhtar, Warburton, and Xu (2017) created a computer-based teaching system that was employed. Findings from Akhtar et al. suggest embedded predictive analytics to target timely learning interventions could improve class performance. Faulconer, Geissler, Majewski, and Trifilo (2013) found that a campus-wide early alert system "has the potential to impact student success by enhancing in real time the lines of communication among student, instructor, and advisor" (p. 47).

Methodology

The steps in this current research project were:

1. *Data collection* – Data collection for the pilot project entailing collected, cleaning and coding CS101 student records from the instructor's gradebooks. The research team then cleaned and organized the data according to the corresponding assignment across the semesters. For example, the programming assignments, problem sets, and exams were organized across all semesters to create a single compiled gradebook. Authors obtained data from historic archives saved in the instructor's gradebooks for the past seven years. Authors then used these data in the creation of the neural networks. A student's record was included in the dataset only if they had a recorded outcome at the end of the semester (i.e. a letter grade or a W for withdrawal). All students were enrolled in CS101 at the authors' home institution. Demographically, the student records were approximately evenly divided in regard to gender, with 52% female and 48% male. Additionally, approximately 40% of the students enrolled were Native American. Regarding the program of study, approximately 90% of the students were majoring in one of the STEM fields. In the end, a total of 592 student records were compiled into the final dataset to be used for neural network training and testing purposes.
2. *Neural network type identification* – Numerous neural network topologies exist and can perform differently given a specific dataset. This research tested 25 different network topologies.

3. *Neural network refinement* – Once a neural network topology has been determined, incremental improvements in accuracy can be realized via refinements.

    1. *Backward elimination* – The researchers first pruned the input space via backword elimination. Backward elimination involves removing a single predictor/factor, rebuilding the neural network and retesting to determine if an improvement in accuracy is realized. If the network's accuracy improves with factor omission, then the factor is removed from the input space. The goal, in this situation, is to have predictive models with only inputs that improve predictive accuracy. By having fewer predictors, a model is less prone to noise within the data and is more generalizable in a production setting.
    2. *Threshold determination* – Once a neural network with a high predictive accuracy has been identified, the threshold for determining whether a student is "at-risk" or not "at-risk" can be varied as a means to find an acceptable balance between Type I and Type II errors. For example, if a threshold of 0.5 is used where the network output is less than 0.5 is interpreted as "at-risk." Then, a threshold value of approximately 0.5 can be tested to see how the overall neural network accuracy responds.
    3. *Sensitivity analysis* – Finally, at the point where a neural network with an acceptable balance between false positives and false negatives has been found, the researchers performed a sensitivity analysis to identify the most impactful predictors. Sensitivity analysis involves varying each predictor/for by a given number of standard deviations and examining how the neural network output responds.
4. *Pilot Experiment* – The researchers piloted the final neural network in a pilot study during the Spring 2021 academic semester.

The results from each of these steps are summarized in the next section.

Results

Data Collection

The first step in developing a predictive system via supervised learning is the acquisition of data to be used for neural network training and testing. For this project, mentors' course grade books for the past seven academic years were collected and compiled. The mentor teaches five sections of Introduction to Computer Programming I each academic year. After cleaning and coding the data, the authors collected 828 complete rows of student data. A significant amount of time and care was spent aligning assignments (i.e., course topics) from one semester to the next and from one academic year to the next. In all, the authors found 12 graded items common across all course sections. Authors deemed the data both reliable and valid. Regarding reliability, the mentor of this research was the only person:

1. Who graded the 12 graded items

3. The only instructor for all course sections

In addition,

1. The same grading scale and assignment weighting were used for all seven years.
2. The same textbook (NOTE: several new editions were released, but no significant change was made to course content)

Neural Network Type Identification

This project used NeuroSolutions Professional by NeuroDimensional to construct and test neural networks. Forty-two neural network architectures were tested, with the PNN performing best on an out-of-sample dataset of 207 rows of student data. The PNN correctly identified 91.3% of the test data (see summary in Table 1). The top 25 performing neural networks and their corresponding accuracy on the out-of-sample dataset are listed in Table 2.

Neural Network Refinement

Backwards Elimination. The number of inputs was refined/reduced using backwards elimination where each input was withheld to determine if the predictive accuracy improved with its inclusion into the predictive system. The goal of backwards elimination is to have only inputs that add to the final predictive accuracy, thereby increasing the generalizability of the final predictive system. After backwards elimination, the input space consisted of twelve inputs.

The initial neural network included all graded items across the entire semester for– Fundamentals of Computer Programming I course for a total of 15 inputs. After backwards elimination, the final neural network had 12 inputs, with the second bookwork assignment and the third exam being trimmed from the input space. The final PNN had the inputs summarized by Table 3.

The resulting neural network had an overall accuracy of 90.8% and is summarized in Table 3. The predictive accuracy was less than the accuracy of the original neural network type identification (91.3%). A neural network with fewer inputs is likely to be more generalizable in a production setting, thereby performing better with new, unseen data.

Threshold Determination. The most impactful incremental improvement occurred by adjusting the threshold of the neural network to a point that maximizes the area under the receiver operating characteristic (ROC) curve. The default threshold value is 0.5 for the PNN output. In other words, a neural network output greater than 0.5 would be interpreted as a student predicted likely to succeed. A network output less than 0.5 would be interpreted as a student who will not be likely to succeed.

The threshold maximizing the area under the ROC curve is shown in Figure 1. Thus, the use of a threshold of 0.51 resulted in a sizable increase in predictive accuracy to 99.2%. At this point, this last refinement resulted in the final neural network summarized in Table 5.

Sensitivity Analysis. To create an early alert system, checkpoints/triggers need to be established at which point students should be contacted regarding their progress in the course. The authors conducted a sensitivity analysis to determine possible checkpoints across the 16-week course. Sensitivity analysis entails varying each neural network input by plus and minus two standard deviations about the mean and measuring the resulting output change in the current PNN across 50 steps on each side of the mean. The outcomes of the sensitivity analysis are detailed in Figure 2.

Fortuitously, three of the top five graded inputs, in regards to sensitivity, occur within the first three weeks of class. Three weeks into the semester should allow an instructor sufficient time to individually help the identified at-risk students change their predicted course. Given the timing of these three inputs, Bookwork 1, the MadLib program and the Property Tax Program, these assignments most likely set the tone of the course for the students. If a student has initial success in their first programming endeavor, then this trend is more likely to continue. The authors hypothesize that if an instructor focuses heavily on students' initial success on the first two programming assignments in COSC 118, then a sizable increase in student success can be realized. With an 8.4 percentage point increase in accuracy with the slight modification of the PNN's threshold from 0.50 to 0.51, this suggests many students are on the cusp of being successful. The authors believe that a focused effort to increase student performance on the first couple of programming assignments could result in a sizable increase in the student success rates for introductory programming courses.

By examining more closely the sensitivity analysis for the three most impactful factors: the MadLib Program, Property Tax Program, and Exam 2, one can see how various scores on these items change the NN output. These relationships are depicted graphically in Figure 3. The sensitivity graphs for all three inputs have sigmoid "S"-shaped curves, suggesting that whatever slight increases in these three assignments can be made, then a corresponding incremental increase in neural network output will result.

These findings suggest that beginning computer science students could benefit greatly by having initial success in their programming efforts. Making struggling students aware of the schools' student success resources relating to programming early in the semester (i.e., tutoring, office hours, open lab time, etc.) could dramatically positively impact student outcomes.

Pilot Study

Pilot Intervention. During the fall 2020 semester, in an effort to assess the effectiveness of the early alert system, the first author of this research used the first three graded items as triggers for interventions taken by the instructor to assist students in their coursework. The three triggers, Bookwork 1, the MadLib program and the Property Tax Program, were all completed and graded within the first three weeks of class.

The instructor began the semester by telling the students about the paramount importance of beginning the semester with a strong start by making perfect submissions for the first couple of

ized and finally demonstrated the use of the

posted rubrics in Canvas to ensure that the students understood how their program would be graded. Then, if a student failed to submit one of the three assignments, the instructor individually contacted the student via email and then by phone if unsuccessful via email reminding the student of the impact of not submitting one of these assignments could have on her or his course outcome. The instructor also sent similar emails to students who did poorly on any of the trigger assignments reminding the student on the use of rubrics and the need to submit complete work to optimize their final course grade.

Pilot Study Results. The student outcomes from fall 2020 are compared to the outcomes from the fall 2019 semester. It should be mentioned upfront that the 2020 semester fundamentally differed from the 2019 semester due to COVID-19. In response to the pandemic, the instructor opted to offer the sections of CS101 in a live online format where the class met via Zoom twice a week during the regularly scheduled class time. Fall 2020 marked the first time teaching online for the instructor and the first time for CS101 at the school to be offered online. Albeit, the instructor recently and fortuitously completed a Quality Matters course entitled "*Improving Your Online Course*" in anticipation of the need to move his courses online. Given the situation, one would reasonably expect the course success rate to drop precipitously for fall 2020. The opposite, however, occurred. Student success rates actually increased, as detailed in the 2X2 contingency table shown in Table 6. A chi-square test of independence showed that there was no significant association between academic semester and course outcome, X^2 (1, N = 93) = 0.62, p = 0.43. The lack of statistical significance ($p < 0.05$) may be attributable to sample size, the minimal treatment undertaken, or to the extraordinary learning environment resulting from being a student during the COVID-19 pandemic.

While the pilot study did not yield a statistically significant result, the outcome of the pilot study suggests that the treatment may be effective. The DFW rate dropped from 31.8% in fall 2019 to 24.5% in 2020. The 23% increase in student success and a 7.3 percentage point decrease in the DFW rate support the continued use of the system. Additionally, 83% of the students who had a DFW outcome in fall 2020 had at least one of the three triggers not submitted, confirming the validity of the early alert system and the identified early alert triggers. This provided the instructor with adequate evidence to continue the early alert system for spring 2021. In doing so, the instructor attained a 16.7% DFW rate for Spring 2021.

Further Research

This paper details the creation of a highly accurate predictive system for identifying at-risk students. Hopefully, an increase in student success rates will be realized. Further research will be needed to determine the most appropriate/successful interventions that will work for students at the authors' home institution. Other institutions of higher education wanting to create their own predictive system will need to do so using a similar methodology but with the data from their introductory computer science courses.

This research could also be treated as a general framework for identifying academic, early alert triggers for other disciplines.

Conclusion

This research demonstrated the ability of a neural network-based predictive system to accurately identify students who were at risk of not succeeding in the introductory programming class at a two-year public institution of higher education. More specifically, a probabilistic neural network accurately classified 99% of students in an out-of-sample test dataset of 207 students.

The authors view this research as a first step to increasing student success rates in introductory CS courses at 2-year public colleges. This article can serve as a framework for other early alert systems for other gateway courses. The next step is to explore treatment options and determine their efficacy. The first attempt at treatment options was piloted by the mentor of this study in Fall 2020. While the pilot study did not yield statistically significant results, the study provided sufficient evidence for the mentor to continue the early alert system's use.

The ultimate goal of this research is to increase student success rates in introductory CS courses, thereby increasing the number of degrees and certificates awarded. With over a third of beginning CS students being stopped by the first course in the program of study, it is incumbent upon computer science educators to find solutions to help all students interested in computer science be successful. Given the tremendous shortage of qualified information technology professionals both nationally and globally, these efforts can definitely result in a positive social change by helping students who have already expressed an interest in computer science to be successful.

Declarations

Availability of data and material:

The data that support the findings of this study are available on request.

Funding

Not applicable

Acknowledgments:

Not applicable

Compliance with Ethical Standards

Ethical Statement. The authors declare that they have no conflicts of interest. All procedures performed in this study involving human participants were in accordance with the ethical standards of the institutional

and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Consent Statement. Informed consent was obtained from all individual participants included in the study.

References

100 Best Jobs of 2020. (n.d.). Retrieved November 29, 2020, from https://money.usnews.com/careers/best-jobs/rankings/the-100-best-jobs

Akhtar, S., Warburton, S., & Xu, W. (2017). The use of an online learning and teaching system for monitoring computer aided design student participation and predicting student success. *International Journal of Technology & Design Education, 27*, 251–270. doi:10.1007/s10798-015-9346-8

Bennedsen, Jens & Caspersen, Michael. (2007). Failure rates in introductory programming. *SIGCSE Bulletin*. 39. 32-36. 10.1145/1272848.1272879.

Chao, H., Qingyu, Y., Mingwei, D., & Donghui, Y. (2017). Financial distress prediction using SVM ensemble based on earnings manipulation and fuzzy integral. *Intelligent Data Analysis, 21*, 617–636. doi:10.3233/IDA-160034

Chaudhuri, A. (2014). Modified fuzzy support vector machine for credit approval classification. *AI Communications, 27*, 189–211. doi:10.3233/AIC-140597

Dalton, D., Moore, C. A., & Whittaker, R. (2009). First-generation, low-income students. *New England Journal of Higher Education, 23*(5), 26–27. Retrieved from http://www.nebhe.org/thejournal/

de Carvalho Filho, A. O., Silva, A. C., de Paiva, A. C., Nunes, R. A., Gattass, M., & Gattass, M. (2017). Computer-aided diagnosis system for lung nodules based on computed tomography using shape analysis, a genetic algorithm, and SVM. *Medical & Biological Engineering & Computing, 55*, 1129–1146. doi:10.1007/s11517-016-1577-7.

Faulconer, J., Geissler, J., Majewski, D., & Trifilo, J. (2013). Adoption of an early-alert system to support university student success. *Delta Kappa Gamma Bulletin, 80*(2), 45–48. Retrieved from http://www.dkg.org/

Hardgrave, B. C., & Wilson, R. L. (1994). Predicting graduate student success: A comparison of neural networks and traditional techniques. *Computers & Operations Research, 21*, 249–264. doi:10.1016/0305-0548(94)90088-4

Hanover Research. (2014). Early Alert Systems in Higher Education. Retrieved November 29, 2020, from https://www.hanoverresearch.com/wp-content/uploads/2017/08/Early-Alert-Systems-in-Higher-Education.pdf.

Hamman, K. (2016). Factors that contribute to the likeliness of academic recovery. *Journal of College Student Retention: Research, Theory & Practice, 20*, 162–175. doi:10.1177/1521025116652636.

Kaur, A., Sood, N., Aggarwal, N., Vij, D., & Sachdeva, B. (2017). Traffic state detection using smartphone based acoustic sensing. *Journal of Intelligent & Fuzzy Systems, 32*, 3159–3166. doi:10.3233/JIFS-169259.

Khan, S. A., Riaz, N., Akram, S., & Latif, S. (2015). Facial expression recognition using computationally intelligent techniques. *Journal of Intelligent & Fuzzy Systems, 28*, 2881–2887. doi:10.3233/IFS-151567.

Kim, S. H., & Hak Chun, S. (1998). Graded forecasting using an array of bipolar predictions: Application of Probabilistic Neural Networks to a stock market index. International Journal of Forecasting, 14(3), 323–337. https://doi.org/10.1016/s0169-2070(98)00003-x

Millea, M., Wills, R., Elder, A., & Molina, D. (2018). What matters in college student success? Determinants of college retention and graduation rates. *Education, 138*, 309–322.

Naik, B., & Ragothaman, S. (2004). Using neural networks to predict MBA student success. *College Student Journal, 38*, 143–150. Retrieved from http://www.projectinnovation.com/college-student-journal.html

NeuroSolutions. (2014). What is Leave-N-Out? Retrieved from http://www.neurosolutions.com /infinity/help/index.html?WhatisLeaveNOut.html

Noel Levitz. (2015). Student Retention and College Completion Practices Benchmark Report for Two-Year and Four-Year Institutions. Retrieved November 29, 2020, from http://learn.ruffalonl.com/rs/395-EOG-977/images/2015RetentionPracticesBenchmarkReport.pdf

Software Developers: Occupational Outlook Handbook. (2020, September 01). Retrieved November 29, 2020, from https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm

Tseng, G. (2007). Support vector machines. In N. J. Salkind (Ed.), *Encyclopedia of measurement and statistics* (Vol. 1, pp. 979–980). Thousand Oaks, CA: Sage.

U.S. Bureau of Labor Statistics. (2021, September 15). Home: Occupational outlook handbook. U.S. Bureau of Labor Statistics. Retrieved September 27, 2021, from https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htmv.

US News and World Report. (n.d.). U.S. news & World REPORT: NEWS, rankings and analysis. 100 Best Jobs. Retrieved September 27, 2021, from https://www.usnews.com/.

Watson C. & Li Frederick Li. (2014). Failure rates in introductory programming revisited. *Proceedings of the 2014 conference on Innovation & technology in computer science education (ITiCSE '14)*. Association

van Heerden, B., Aldrich, C., & du Plessis, A. (2008). Predicting student performance using artificial neural network analysis. *Medical Education, 42*, 516–517. doi:10.1111/j.1365-2923.2008.03052.x

Vapnik, V. (1995). *The nature of statistical learning theory*. New York, NY: Springer-Verlag.

Yang, Z. R., Platt, M. B., & Platt, H. D. (1999). Probabilistic neural networks in bankruptcy prediction. Journal of Business Research, 44(2), 67–74. https://doi.org/10.1016/s0148-2963(97)00242-7

Yang, Z. R., Platt, M. B., & Platt, H. D. (1999). Probabilistic neural networks in bankruptcy prediction. Journal of Business Research, 44(2), 67–74. https://doi.org/10.1016/s0148-2963(97)00242-7

Tables

Table 1
Most Accurate Network

Most accurate network (threshold = 0.50)		Predicted	
		At-risk	Not at-risk
Actual	At-risk	53	10
	Not at-risk	8	136
Overall accuracy (53+136)/207 = 91.30%			

Table 2
Top 25 Neural Networks

Rank	Model Name	Correct
1	**PNN-0-N-N (Probabilistic Neural Network)**	**91.30%**
2	MLPR-1-B-R (Regression MLP)	90.82%
3	MLPR-2-O-M (Regression MLP)	90.34%
4	MLPR-2-B-R (Regression MLP)	90.34%
5	MLPC-1-B-R (Classification MLP)	89.86%
6	GFFR-1-B-R (Reg Gen Feedforward)	89.86%
7	MLPR-1-B-L (Regression MLP)	89.37%
8	GFFR-1-B-L (Reg Gen Feedforward)	89.37%
9	MLPC-2-B-L (Classification MLP)	89.37%
10	SVM-0-N-N (Classification SVM)	87.44%
11	LogR-0-B-L (Logistic Regression)	86.96%
12	MLPRPC-1-B-L (Reg MLP with PCA)	86.47%
13	MLPR-1-O-M (Regression MLP)	85.99%
14	MLPC-1-O-M (Classification MLP)	85.99%
15	GFFC-1-B-L (Class Gen Feedforward)	85.99%
16	MLPRPC-1-O-M (Reg MLP with PCA)	85.99%
17	RBF-1-B-R (Radial Basis Function)	85.99%
18	LogR-0-B-R (Logistic Regression)	85.51%
19	RBF-1-B-L (Radial Basis Function)	85.51%
20	MLPCPC-1-O-M (Class MLP with PCA)	85.02%
21	TLRN-1-B-R (Time-Lag Recurrent Network)	84.06%
22	GFFR-1-O-M (Reg Gen Feedforward)	83.57%
23	LinR-0-B-L (Linear Regression)	83.09%
24	MLPC-2-B-R (Classification MLP)	83.09%
25	TDNN-1-B-R (Time-Delay Network)	83.09%

Note. PNN = Probabilistic Neural Network.

Table 3
Neural Network Inputs after Backwards Elimination

Number	Input Name
1	Bookwork 1
2	Madlib Program
3	Property Tax Program
4	Software Sales Program
5	Car Class Program
6	Program 5
7	Bookwork 3
8	TicTacToe Program
9	Exam1
10	Exam2
11	In-Class Final
12	Take-Home Final

Table 4

Overall Accuracy After Backwards Elimination

		Predicted	
(threshold = 0.50)		At-risk	Not at-risk
Actual	At-risk	52	10
	Not at-risk	9	136
Accuracy		85.2%	93.1%
Overall accuracy		(52 + 136)/207 = 90.8%	

Table 5

Final Predictive PNN

Elimination		Predicted	
(threshold = 0.51)		At-risk	Not at-risk
Actual	At-risk	59	0
	Not at-risk	2	146
Accuracy		96.7%	100%
Overall accuracy		(59 + 146)/207 = 99.2%	

Table 6

2 X 2 Contingency Table to Pilot Study

Course Outcome	Academic Semester		χ^2
	Fall 2019	Fall 2020	
Success	30	37	0.62*
DFW	14	12	

Note. * = p = 0.43

Figures

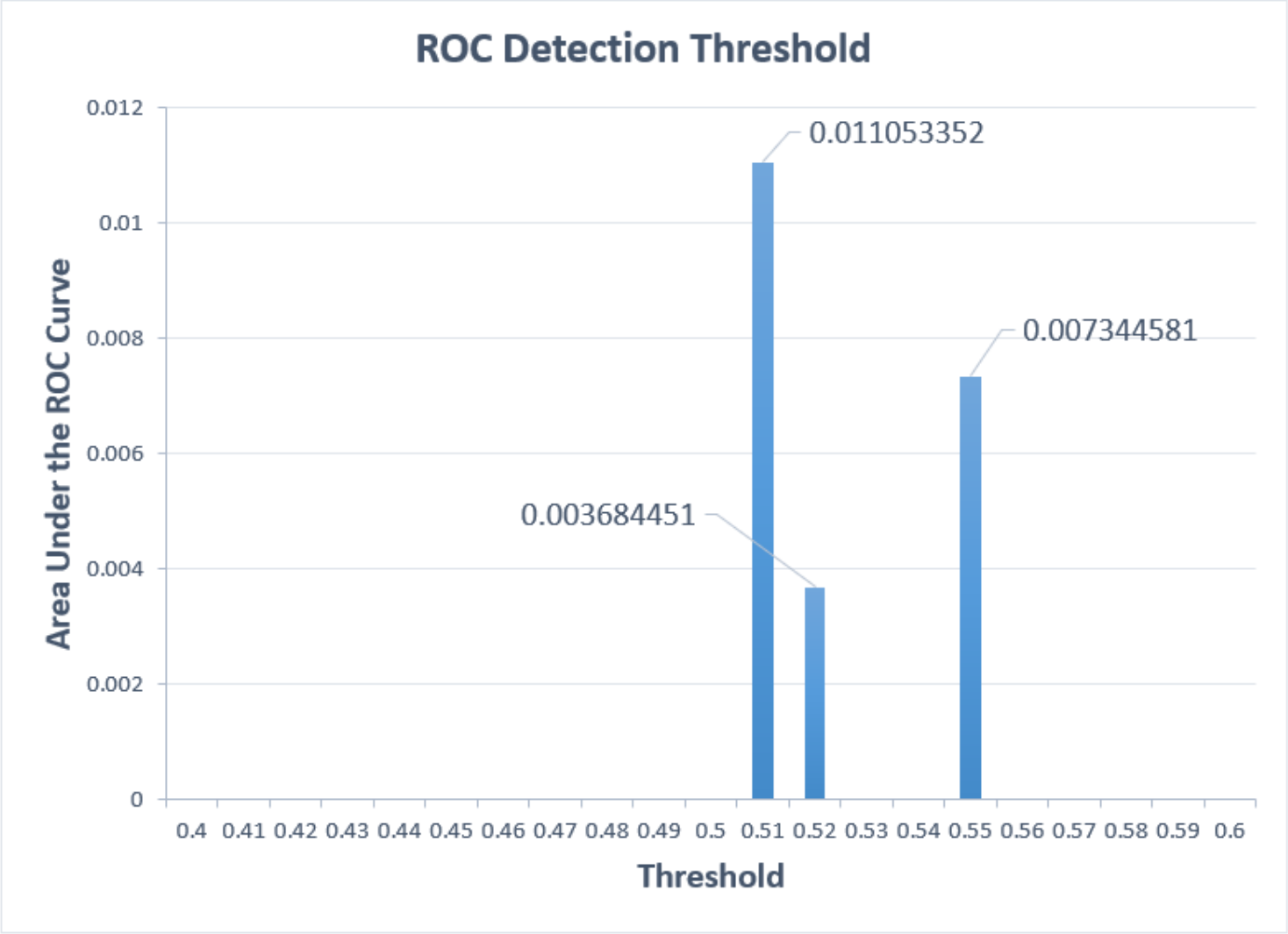

Figure 1

Increase in area under the ROC curve across different thresholds. Note. (N = 592 student records used in threshold calculations).

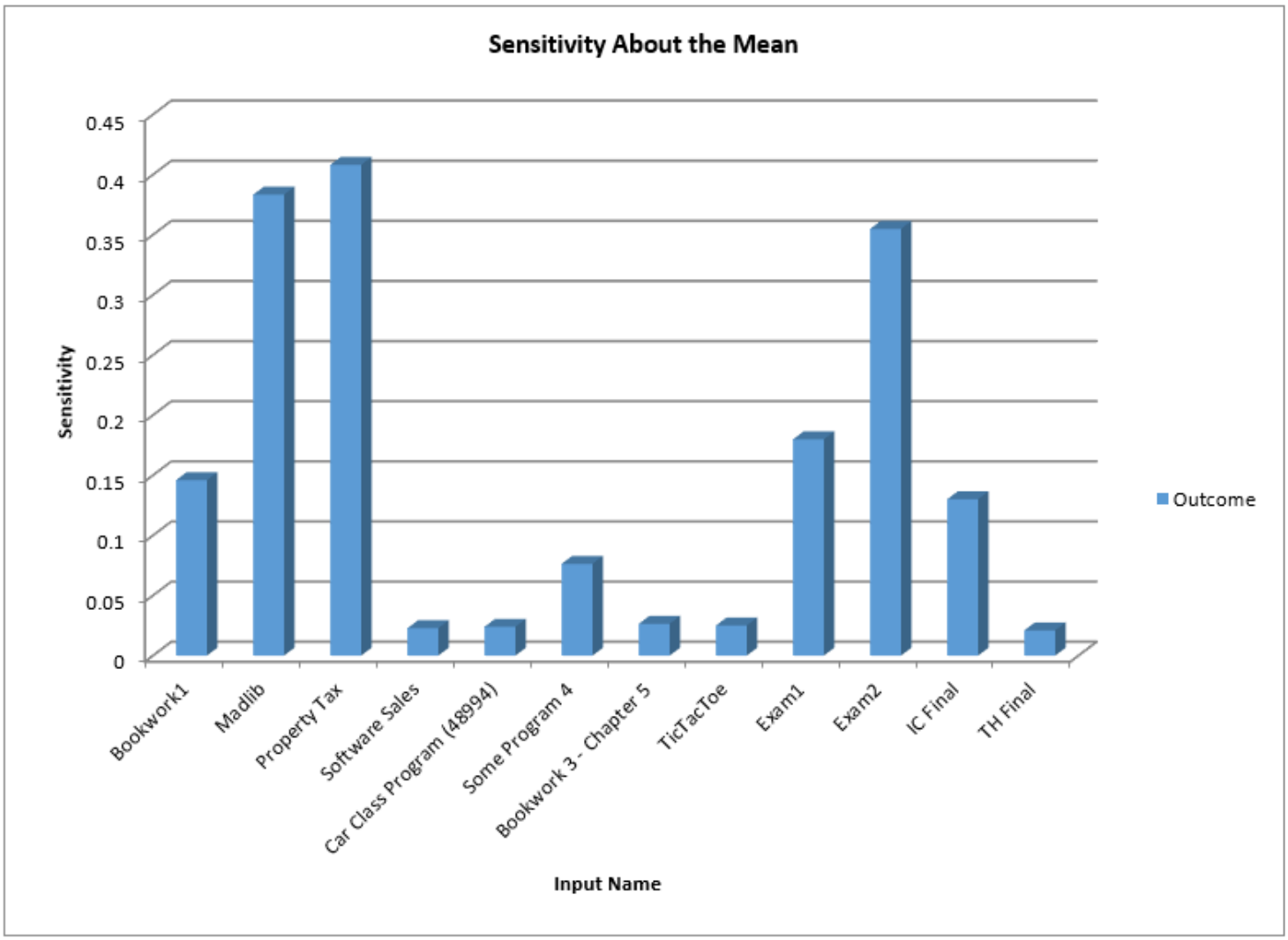

Figure 2

Sensitivity about the Mean. Note. (N = 592 student records used in Sensitivity About the Mean calculations).

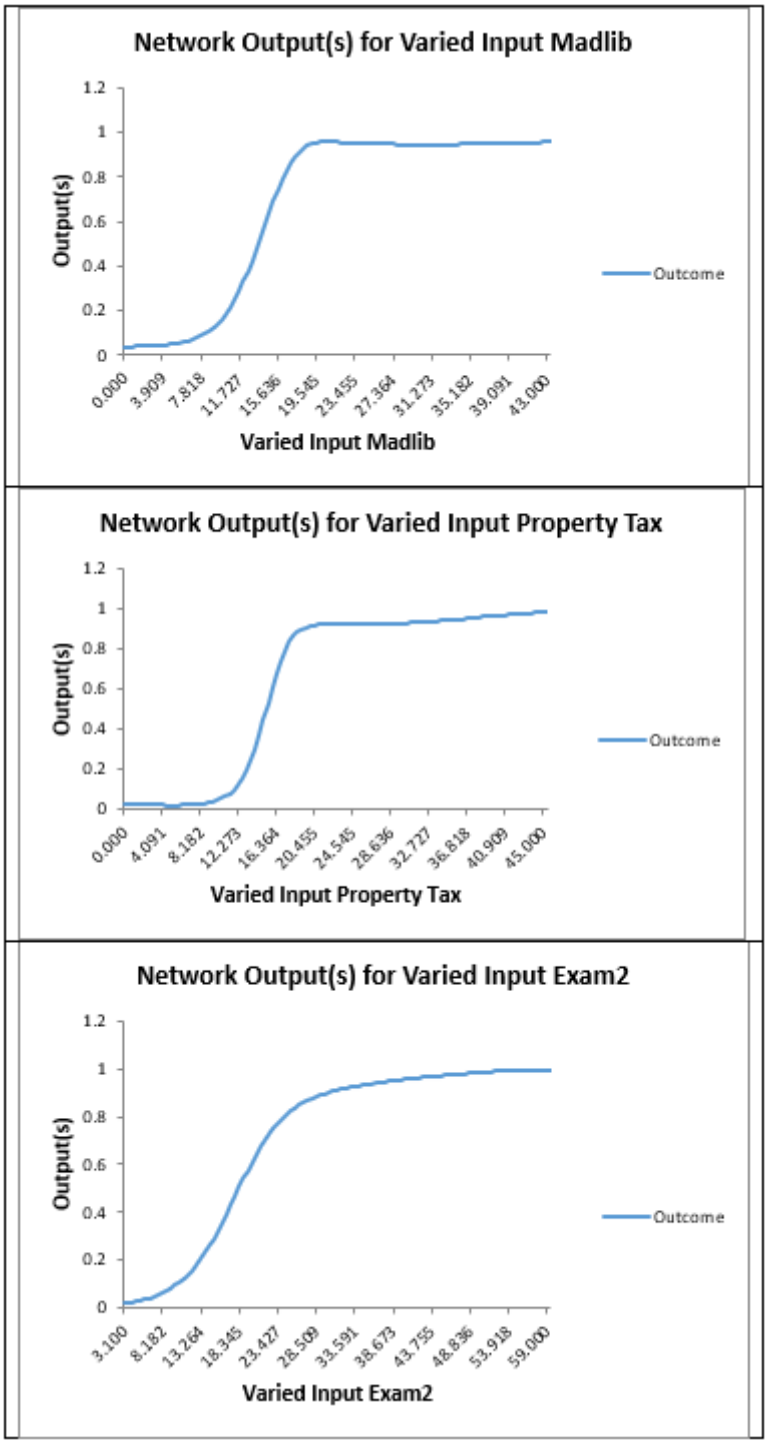

Figure 3

Sensitivity of the mean for the three most impactful factors. Note. (N = 592 student records used in sensitivity calculations).