Advances in Artificial Intelligence and Machine Learning; Research 2 (2) 407-421 Received 10-06-22; Accepted 23-06-22; Published 01-07-22

Using Machine Learning to Identify At-risk Students in an Introductory Programming Course at a Two-year Public College

Cameron Cooper cooperc@sanjuancollege.edu
Computer Science, San Juan College
Farmington, New Mexico,
United States

Corresponding Author: Cameron Cooper

Abstract

Nationally, more than one-third of the students who enroll in an introductory computer science programming course (CS1) do not succeed. To improve student success rates, supervised machine learning is used to identify students who are "at risk" of not succeeding in CS1 at a two-year public college. The resultant predictive model accurately identifies ≈99% of at-risk students in an out-of-sample test dataset. The course 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 CS1 in fall 2020. The outcome of this pilot study was a 23% increase in student success and a 7.3% decrease in the DFW rate (i.e. the percentage of students who receive a D, receive an F, or withdraw). More importantly, this study identified academic-based early alert triggers for CS1. The first two graded programs are of paramount importance for student success in this course.

Keywords: Computer science, Early alert, Early alert triggers, Machine learning, Student success, Neural networks, Gateway course.

1. INTRODUCTION

Bennedsen and Caspersen found student success rates at colleges and universities in CS1to be approximately 67%. Seven years later [1], via meta-analysis, Watson and Li found the student success rates in to be essentially unchanged at 67% [2]. 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 [3]. The author defines 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 provided accurate predictions for students who were unlikely to succeed in CS1. These measures can 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 changing the trajectory of the student toward course success.

Citation: Cameron Cooper Using Machine Learning to Identify At-risk Students in an Introductory Programming Course at a Two-year Public College Advances in Artificial Intelligence and Machine Learning. 2022;2(3):27.

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 the D grade, Fail, Withdraw (DFW) rate and a 23% increase in student success for CS1 at the researchers' home institution.

2. BACKGROUND

According to the College's Strategic Plan Annual in 2020-21, one of the strategic goals of the author's 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 study 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 CS1 at this college historically stands at 61.8%, five percentage points below the national average. As almost 40% of the students who are willing to consider computer science by taking CS1 are 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 [4]. 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) [5]. The opportunity cost for students unable to advance in a field ranked as providing the best jobs according to the U.S. News and World Report is substantial (v). Efforts to improve student success must be undertaken. This study proposes an early alert system in which, 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.

3. LITERATURE REVIEW

Neural networks have been employed as a means to predict student success in numerous contexts dating back to the mid-1990s. Hardgrave and Wilson used neural networks to predict graduate student success [6]. Naik and Ragothaman utilized neural networks to predict MBA student success [7]. 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 demonstrated the ability of neural networks to predict student success in medical school [8].

Hanover Research offers a comprehensive overview of Early Alert Systems in Higher Education [9]. 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 component and contributes to the student success literature by considering how 'alerts' (i.e., triggers) are determined. The author believes that the accuracy of the alert component of an effective system is vital to the system's success. The author employs 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 are 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 [10]. However, only 57.1% of these schools found their systems to be "very or somewhat effective." This study 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" or "postenrollment" factors (p. 11). This study utilizes postenrollment factors (i.e., student performance data on specific graded items in Introductory Computer Science).

Probabilistic neural networks (PNNs), 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 [11], various signal processing applications, plant classification using leaf structures, and bankruptcy prediction [12]. This study demonstrates the applicability of a PNN to accurately predict student success to assist targeting interventions.

Machine learning has been successfully applied to identify at-risk students in previous studies. E. Er, utilized a combination of three machine learning techniques (instance-based learning classifier, decision trees, and naïve Bayes) to accurately predict student success in the field of information systems [13]. S.B. Kotsiantis demonstrated how individual student assignments can be incorporated into the creation of a decision support system for tutors [14]. This study differs from the existing body of research in several aspects:

1. This study demonstrates the applicability of machine learning to predict student success in an introductory programming course.
2. Second, this study demonstrates the applicability of neural networks to predict student success with a high degree of accuracy.
3. Last, the pilot study detailed in this work offers evidence that the identified early alert triggers can be successfully used to increase student success.
4. The findings provide other computer science educators with a framework for the development of their own "at-risk" early alert systems.

An additional outcome from this study is the identification of factors that predict student success in introductory computer science courses. The identification of predictive factors impacting student success has been addressed by multiple researchers. Dalton, Moore et al., studied the impact of being a first-generation and low-income student on student success [15]. Karen Hamman published a study of factors that contribute to academic recovery [16]. Millea, et al., presented factors determining college retention and graduation rates [17].

After the predictive factors are determined and an accurate predictive system is constructed, an early alert system needs to be employed to improve student outcomes. Akhtar, et al., created a computer-based teaching system that employed a computer support collaborative learning environment designed to support lab-based CAD teaching [18]. The findings of Akhtar et al. suggest that embedded predictive analytics to target timely learning interventions could improve class performance [18]. Faulconer, et al., 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) [19].

4. METHODOLOGY

The steps in this research project were as follows:

1. **Data collection** – Data collection for the pilot project entailed the collection, cleaning and coding of CS1 student records from the instructor's gradebooks to create the training and testing dataset for the predictive system. The research team then cleaned and organized the data according to the corresponding assignments across semesters. For example, the programming assignments, problem sets, and exams were organized across all semesters to create a single compiled gradebook. The author obtained data from historic archives saved in the instructor's gradebooks for the past seven years. The author then used these data in the creation of the neural networks. A student's record was included in the dataset only if the student 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 CS1 at the author's 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. This sample size is the maximum number of complete student records and was not based upon any statistical calculation. The goal of training and testing neural networks is to have as much data as possible to provide adequate training to create an accurate neural network.

2. **Neural network type identification** – Numerous neural network topologies exist, and they can perform differently given a specific dataset. The research team utilized NeuroSolutions by nDimensional Neural Network software to create the neural networks tested for this study. NeuroSolutions offers a robust list of neural network topologies, shown in TABLE 2. This study tested 25 different network topologies. The list of 25 different network topologies includes representative neural network topologies from all the major neural network types (i.e., multilayer perceptrons, support vector machine, probabilistic neural network, regression networks and principal component analysis networks).

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

   (a) *Backward elimination* – The researchers first pruned the input space via backward elimination. Backward elimination involves removing a single predictor/factor, rebuilding the neural network and retesting to determine whether 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 obtain 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.

(b) *Threshold determination* – Once a neural network with a high predictive accuracy is 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, then the network output 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.

(c) *Sensitivity analysis* – Finally, when a neural network with an acceptable balance between false positives and false negatives is found, researchers perform a sensitivity analysis to identify the most impactful predictors. Sensitivity analysis involves varying each predictor 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.

5. RESULTS

5.1 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, the mentor's 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 author collected 592 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 author found 12 graded items common across all course sections. The author deemed the data both reliable and valid. Regarding reliability, the mentor of this research was

1. The only person who graded the 12 graded items
2. The only person who entered the data into the grading program, and
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 was used for all seven years (NOTE: several new editions were released, but no significant change was made to course content).

5.2 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 207 rows of data comprised approximately 35% of the student data and randomly selected from the 592 rows of data as an out-of-sample test dataset. This data was set aside and used to test the models' accuracies once they had been created, a common practice used by data scientists to test how a given model will perform on production data (i.e. the generalizability of the predictive system). 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.

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%			

The best performing neural network on an out-of-sample dataset was a Probabilistic neural network (PNN). PNNs were first conceived by Specht in 1990 [20]. Compared to the most commonly used neural network, a multilayer perceptron (MLP). PNNs essentially replace the sigmoid activation function (e.g. the hyperbolic tangent function) with a statistically derived "exponential function. With this substitution, a probabilistic neural network (PNN) can then compute nonlinear decision boundaries which approach the Bayes optimal is formed." (p. 109), which is a big advantage of a PNN over a MLP. The author believes this advantage is evidenced by the very high accuracy of the final predictive system (99+%) found in this study. The disadvantage of a PNN compared to a MLP is a PNN requires much more memory in operation and may need additional hardware to support their continued use. This is dependent upon the size of the network. In regards to this research, this was not an issue due to the small size of the neural network (n = 592).

5.3 Neural Network Refinement

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

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.

The initial neural network included all graded items across the entire semester for the Fundamentals of Computer Programming I course, for a total of 15 inputs. After backward elimination, the final neural network had 12 inputs, with the second bookwork assignment and the third exam being trimmed from the input space. The inputs of the final PNN are summarized in 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 is not likely to succeed, TABLE 4.

Table 3: Neural Network Inputs After Backward 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 Backward Elimination

(threshold = 0.50)		**Predicted**	
		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%			

The threshold maximizing the area under the ROC curve is shown in FIGURE 1. The use of a threshold of 0.51 resulted in a sizable increase in predictive accuracy to 99.2%. This last refinement resulted in the final neural network, summarized in TABLE 5.

Table 5: Final Predictive PNN

Elimination (threshold = 0.51)		**Predicted**	
		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%			

Sensitivity Analysis. To create an early alert system, checkpoints/triggers need to be established at which students should be contacted regarding their progress in the course. The author 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 described in FIGURE 2.

Fortuitously, three of the top five graded inputs, with regard to sensitivity, occur within the first three weeks of class. Three weeks into the semester should allow an instructor sufficient time to

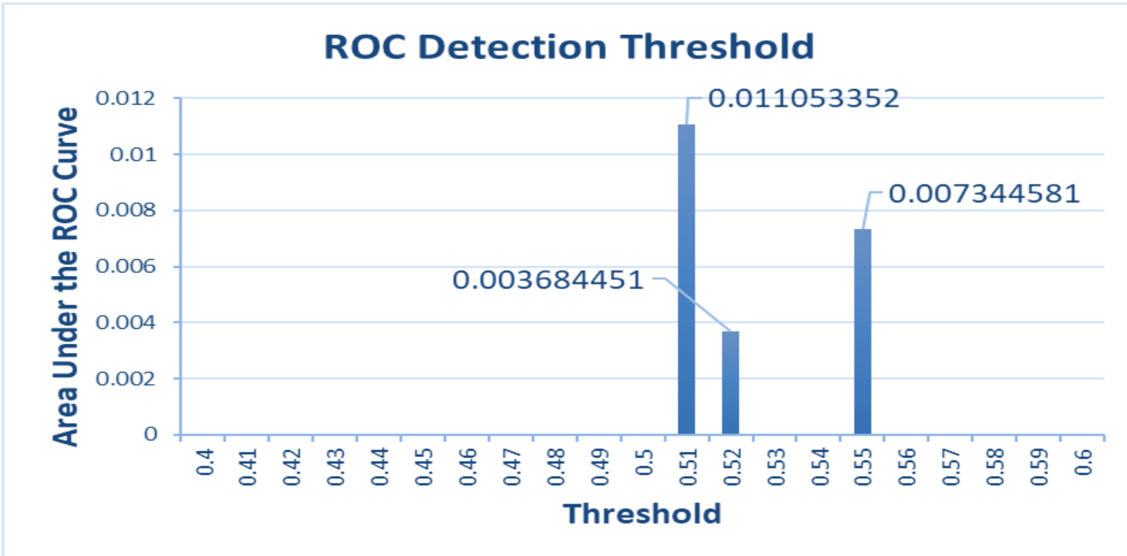

Figure 1: Increase in area under the ROC curve across different thresholds.
Note. (N = 592 student records used in threshold calculations).

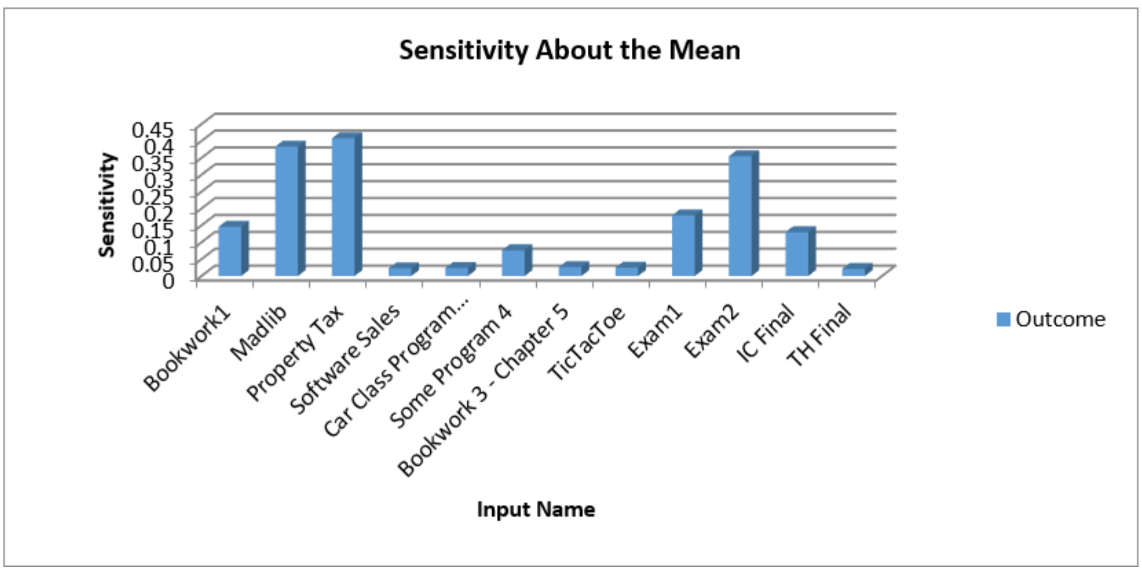

Figure 2: Sensitivity about the Mean.
Note. (N = 592 student records used in sensitivity about the mean calculations).

individually help the identified at-risk students change their predicted course. Given the timing of these three inputs, the Bookwork 1, MadLib program and Property Tax Program assignments most likely set the tone of the course for the students. If a student has initial success in her first programming endeavor, then this trend is more likely to continue. The author hypothesizes that if an instructor focuses heavily on students' success in the first two programming assignments in COSC 118, then a sizable increase in student success can be realized. The 8.4 percentage point increase in accuracy accompanying the slight modification of the PNN's threshold from 0.50 to 0.51 suggests that many students are on the cusp of being successful. The author believe that a focused effort to enhance student performance on the first couple of programming assignments could result in a sizable increase in the student success rates for introductory programming courses.

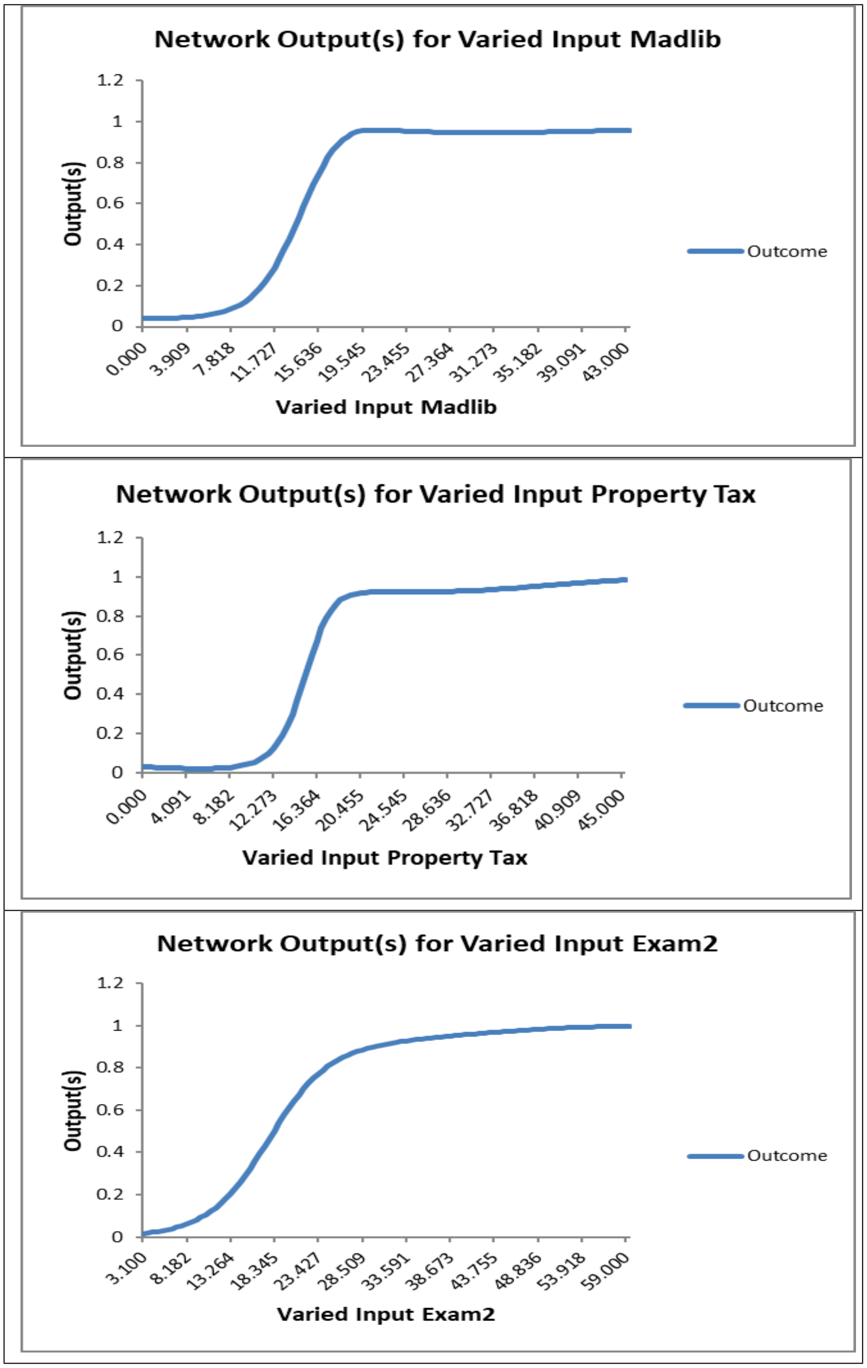

Figure 3: Sensitivity of the mean for the three most impactful factors.
Note. (N = 592 student records used in sensitivity calculations)

By more closely examining 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 neural network 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 (e.g., tutoring, office hours, open lab time) could have a dramatic, positive impact student outcomes.

5.4 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 study 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 programs. The instructor repeatedly and strongly emphasized 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, if no response was received, then by phone to remind the student of the impact not submitting one of these assignments could have on her or his course outcome. The instructor sent similar emails to students who did poorly on any of the trigger assignments, reminding the students about 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 CS1 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 the instructor taught online and the first time CS1 was offered online at the school. However, the instructor had recently 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. Using a freely available online 2X2 contingency table calculator from Vassarstats.net (http://vassarstats.net/tab2x2.html), 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.

Table 6: 2 X 2 Contingency Table to Pilot Study

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

Note. * = p = 0.43

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 the 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 not submitted at least one of the three triggers, 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 and fall 2021. By doing so, the instructor attained a 16.7% DFW rate for spring 2021 and a 15.6% rate for fall 2021.

Study Limitations This research relies heavily on a stable curriculum for multiple semesters in order to have enough data collected to train and test an accurate neural network. If a computer science department makes any significant curricular change, then a new predictive system will need to be created, which could take at least an academic year in order to collect a sufficient amount of data.

5.5 Further Research

This paper describes 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 is needed to determine the most appropriate/successful interventions that will work for students at the author's 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 own introductory computer science courses. Additional research needs to be performed to determine the applicability of this study to other fields of study (i.e., other gateway courses with high DFW rates).

Ideally, this study could be treated as a general framework for identifying academic, early alert triggers for other disciplines.

6. CONCLUSION

This study 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. A probabilistic neural network was used to accurately classify 99% of students in an out-of-sample test dataset of 207 students. While other neural network and deep learning methods exist, the author believes the 99% accuracy obtained using a probabilistic neural network is sufficient for the purposes of this research in identifying early alert triggers for introductory computer science (CS1) instructors to use to target their academic success interventions.

The most significant finding of this research is the mission-critical importance the first couple of graded programs have on the ultimate success for students in CS1. Logistically, CS1 students must surmount a sizable learning curve to be able to successfully submit their first programming assignment. They need to:

1. Learn how to enter their code into an IDE (Integrated Development Environment).
2. Compile their code.
3. Understand any compilation errors.
4. Fix their errors and rerun their code.
5. Interpret the output.
6. Repeat steps 2. Through 5. until they have a working version of their programming assignment without any compilation and runtime errors.
7. Make a submission to their instructor following the submission procedure for the course.
8. Understand each of the previous steps without any prior programming experience.

Being able to successfully navigate these steps to write and submit their first programs, CS1 students will be in a much better situation for further success as the semester progresses. This is again evidenced by this study's results.

The author views this study as a first step to increasing student success rates in introductory CS1 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 author 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 using the early alert system.

The ultimate goal of this study is to increase student success rates in introductory CS courses, thereby increasing the number of degrees and certificates awarded. As over one-third of beginning CS students are 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 positive social change by helping students who have already expressed an interest in computer science to be successful.

7. AVAILABILITY OF DATA AND MATERIAL

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

8. FUNDING

Not applicable

9. ACKNOWLEDGMENT

Not applicable

10. COMPLIANCE WITH ETHICAL STANDARDS

Ethical Statement. The author declares that he has 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.

11. CONSENT STATEMENT

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

References

[1] Jens B, Michael C. Failure Rates in Introductory Programming. SIGCSE Bulletin. 2007;39:32-36.

[2] Watson C, Li LF. Failure Rates in Introductory Programming Revisited. Proceedings of the 2014 Conference on Innovation & technology in computer science education (ITiCSE '14). Association for Computing Machinery, New York, NY, USA. 2014;39–44.

[3] Hertz M. What Do "CS1" and "CS2" Mean?: Investigating Differences in the Early Courses. SIGCSE '10: Proceedings of the 41st ACM technical symposium on Computer science education. 2010;199–203.

[4] https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm

[5] https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htmv

[6] Hardgrave BC, Wilson RL. Predicting Graduate Student Success: A Comparison of Neural Networks and Traditional Techniques. Computers & Operations Research. 1994;21:249-264.

[7] Naik B, Ragothaman S. Using Neural Networks to Predict MBA Student Success. College Student Journal. 2004;38:143–150. http://www.projectinnovation.com/college-student-journal.html

[8] van Heerden B, Aldrich C, du Plessis A. Predicting Student Performance Using Artificial Neural Network Analysis. Medical Education. 2008; 42, 516–517.

[9] https://www.hanoverresearch.com/wp-content/uploads/2017/08/Early-Alert-Systems-in-Higher-Education.pdf.

[10] http://learn.ruffalonl.com/rs/395-EOG-977/images/2015RetentionPracticesBenchmarkReport.pdf

[11] Kim S H, Chun SH. Graded Forecasting Using an Array of Bipolar Predictions: Application of Probabilistic Neural Networks to a Stock Market Index. International Journal of Forecasting. 1998;14:323-337.

[12] Yang ZR, Platt MB, Platt HD. Probabilistic Neural Networks in Bankruptcy Prediction. Journal of Business Research. 1999;44:67-74.

[13] Er E. Identifying At-Risk Students Using Machine Learning Techniques: A Case Study With Is 100. International Journal of Machine Learning and Computing. 2012;2: 476-480.

[14] Kotsiantis SB. Use of Machine Learning Techniques for Educational Proposes: A Decision Support System for Forecasting Students' Grades. Artif Intell Rev. 2012;37:331-344.

[15] Dalton D, Moore CA, Whittaker R. First-Generation, Low-Income Students. New England Journal of Higher Education. 2009;23:26-27.

[16] Hamman K. Factors That Contribute to the Likeliness of Academic Recovery. Journal of College Student Retention: Research, Theory & Practice. 2016;20:162-175.

[17] Millea M, Wills R, Elder A, Molina D. What Matters in College Student Success? Determinants of College Retention and Graduation Rates. Education. 2018;138, 309–322.

[18] Akhtar S, Warburton S, Xu W. 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. 2017;27:251-270.

[19] Faulconer J, Geissler J, Majewski D, Trifilo J. Adoption of an Early-Alert System to Support University Student Success. Delta Kappa Gamma Bulletin. 2013; 80:45-48.

[20] Specht DF. Probabilistic Neural Networks. Neural Networks. 1990;3:109-118.