Predicting Success for Computer Science Students in CS2 using Grades in Previous Courses

Sulav Malla, Jing Wang, William Hendrix, Ken Christensen
Department of Computer Science and Engineering
University of South Florida, Tampa, USA
sulavmalla@mail.usf.edu, jingwang@cse.usf.edu, whendrix@usf.edu, christen@cse.usf.edu

***Abstract*—In this Work in Progress Innovative Practice paper, we describe a process for finding predictors for student success – and failure – for Computer Science and Computer Engineering students with a focus on the second programming course (CS2). We use readily available off-the-shelf statistical and data mining tools for generating summary statistics, calculating correlations, testing statistical significance, and creating decision trees. We analyze grade data from the first programming course (CS1), entry-level STEM courses (Calculus and Physics), and an English course to determine success predictors for CS2. Not surprisingly, the grade in CS1 is the best predictor for success in CS2. We also find that success in CS2 is independent of gender. Looking deeper into the data, we find characteristics of students who are very likely to pass or fail CS2. Being able to identify predictors for success is useful for calibrating admission criteria and designing appropriate interventions (e.g., requiring pre-req classes, recitation sessions, and so on) to improve success probability for all students. A key contribution of this paper is a step-by-step process that can be used by other programs to find success predictors and design appropriate interventions.**

***Index Terms*—Admission GPA, CS1, CS2, Data mining.**

I. Introduction

The Department of Computer Science and Engineering at the University of South Florida has four undergraduate programs and a graduate program offering MS and PhD. The Department has 28 tenure-track faculty and 11 full-time instructors. About 340 students per year graduate from the four undergraduate programs, which are Computer Science, Computer Engineering, Information Technology, and Cybersecurity. The first two programs are Calculus-based and success of students taking the second programming course (CS2) in these two programs is the context of this paper. Admission into the Computer Science and Computer Engineering programs requires a minimum overall GPA of 3.1 in six entry-level courses: Calculus 1 and 2, Physics with Lab 1 and 2, and English Composition 1 and 2, abbreviated as Calc1/2, Phys1/2, and Comp1/2 respectively. We refer to this admission GPA in six entry level courses as "AdmitGPA" throughout the paper. In addition, a minimum grade of 'B' is required in the first programming course (CS1) for admission. For continuation in the program, a minimum grade of 'B' is required in CS2, that is, students will "pass" CS2, if they have a 'B' or higher and "fail" otherwise. The problem is that about one-third of students fail CS2 in any given semester.

The CS1 and CS2 courses in the department teach programming skills to our students. CS1 uses Java, then CS2 uses C programming language. Following CS2 is CS3 – a course focused on objected oriented program design using C++. This leads to Data Structures, taught using C++.

In this paper, we use freely available statistical packages in Python and WEKA [1] to analyze the Calculus, Physics, English Compositions, and CS1 grades to determine predictors for success – and failure – in CS2. The goal is to identify at-risk students who are likely to fail CS2 so that interventions can be put into place. Key contributions of this paper are:

- Demonstration of how readily available tools can be used to analyze grade data for predictors of student success
- Key findings that corroborate with the literature [2], [3], that CS1 is the best predictor for success in CS2
- Description of an automated procedure to identify at-risk student groups that are very likely to fail CS2

II. Methodology

In this section, we describe the methods and tools used to analyze grades in seven courses (Calculus 1 and 2, Physics 1 and 2, English 1 and 2, and CS1) to predict student performance in CS2.

A. Methods and Tools Used

A step-by-step procedure to analyze the grade data is given below. Anyone with similar grade data can apply these procedures. We consider grade points in previous courses as input variables and student success (pass/fail) in the target course (CS2) as the output variable.

1) Grade distribution and correlation between courses: The first step is to find the grade distribution in each course by plotting a histogram to get a sense of the data. A scatter plot between pairs of courses allows for an observation of the joint distribution of the grades. We then check for correlation between the target course with previously taken courses by calculating Pearson correlation coefficient. A higher correlation value would suggest that students who do better in a particular course tend to do better in the target course as well.

2) Average grade points and t-test: The average grade point (along with 95% confidence interval) in previous courses for two groups, students who pass and those who fail the target course is calculated. A non-overlapping confidence interval between pass and fail groups would indicate that the averages are statistically significantly different (the converse is not true, that is, overlapping confidence interval does not indicate difference is not significant). Welch's t-test between the grade points of the two groups (pass/fail) in each course confirms if there is any significant difference in mean grade points. A significant difference would suggests that grades in that course are a good predictor of student success in the target course.

3) Course importance ranking: Welch's t-test provides us with courses that are good predictors of student success in the target course, but some courses might be better predictors than others. We would now like to rank the previous courses in terms of their predictability to student success. The correlation coefficient calculated in the first step can rank courses according to their importance, the higher the correlation, the better the course grade in predicting success. However, correlation is just one metric and a better way would be to rank courses using various metrics to look at the average ranking. We can apply different data mining algorithms to rank input variables according to their importance in predicting the output variable. Using WEKA [1], we apply four different attribute ranking algorithms including Chi-Squared, Gain Ratio, Information Gain, and Relief-F as done in previous research [4]. We do not describe each algorithm due to limited space, refer to reference [1] for further details. Each algorithm uses a metric to evaluate the importance of individual input variables in predicting the output variable, thus creating a ranking of courses from best to worst. We look at the average ranking by the algorithms. A course that gets ranked higher on average is better at predicting student success in the target course.

4) Decision tree classifier: Classification algorithms such as Neural Network, Decision Tree, and Naive Bayes have been shown to have good accuracy in predicting student performances [5]. While a neural network is very promising, it requires a large amount of data to train on (most student grade datasets are small) for accurate results. Furthermore, they act like a black-box providing little intuition/explanation on how results are generated. We use decision trees [6] as they are simple and easy to build. Decision tree models are also explainable, often providing intuitive If-Then-Else rules. Using the `scikit-learn` [7] package in Python, a decision tree can be created which predicts whether a student will pass or fail given their grades in previous courses.

B. Data Collection

CS2 is one of the courses which requires a permit from the department to register. Permit is issued based on AdmitGPA, grade in CS1, and availability of seats. In one particular semester, a total of 253 students applied for a permit in CS2, out of which 148 students were granted permission (our study group in this paper). We collected the following data on 148 students who took the CS2 course in the semester under study: gender, major (Computer Science or Computer Engineering), department admission status, grades in six entry level courses, grades in CS1, and grades in CS2. Grade point in a course is between 4 to 0 for grade letter from 'A' to 'F' as standard. Phys1, Phys2, Comp1, and Comp2 are 3 credit hours, Calc1 and Calc2 are 4 credit hours and each Physics course has a lab of 1 credit hour. With this information, AdmitGPA was calculated as the average overall GPA of grades in the six entry level courses. Note that a few students may not have grades for a course because they may be currently taking the course or might apply Advanced Placement credit towards that course. These records were appropriately marked and excluded from calculations. For example, if a student is taking Phys2 in the same semester as CS2, AdmitGPA is based only on the five other courses for which grades are available.

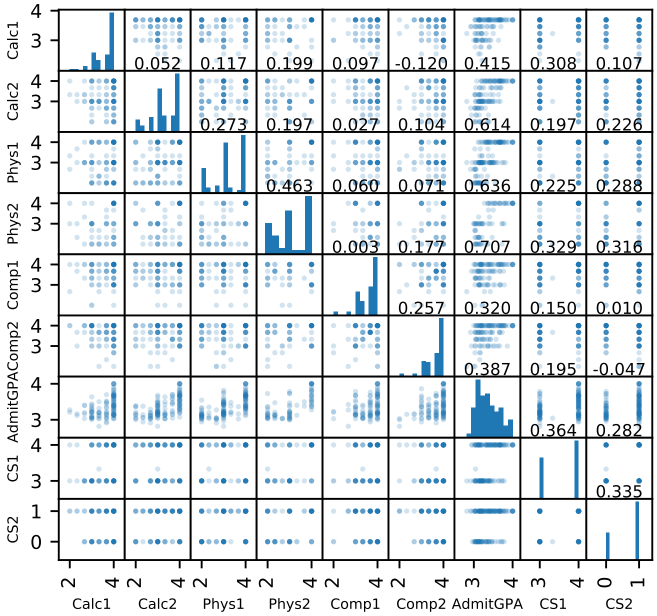

Fig. 1. A scatter plot matrix between grade points in eight courses (Calc1, Calc2, Phys1, Phys2, Comp1, Comp2, AdmitGPA, CS1) and pass (as 1) or fail (as 0) in CS2. The main diagonal has the histogram of grade point distribution and the upper triangle is annotated with Pearson correlation values.

III. Analysis and Results

In this section we present the analysis of student data and the corresponding results using the methods outlined above.

A. Student Success in CS2

Out of the 148 students, 101 (68.2%) successfully passed CS2 with an 'A' or a 'B' while 47 (31.8%) students failed. In the following analysis, grade points in previous courses are the input variables while success in CS2 is the output variable.

1) Histogram, scatter plot, and correlation: In Fig. 1 we plot a scatter plot matrix between AdmitGPA, grade points in six entry-level courses, CS1, and CS2 (pass is denoted as 1 and fail as 0 in CS2). This plot is symmetrical around the main diagonal and the calculated Pearson correlation values are annotated in the upper triangle. The main diagonal shows the histogram of each of the nine courses. The color intensity of points in the scatter plots represent the number of students (the greater the number of students at a point, the darker the point). We are more interested in the last column (and last row) which has correlation values (and scatter plot) between success in CS2 with grade points in other courses. The following observations can be made from this figure:

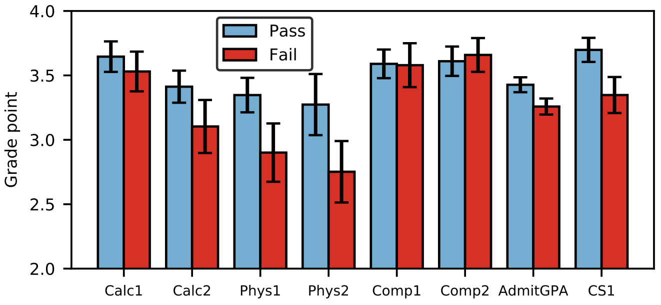

Fig. 2. Average grade points in different courses with 95% confidence interval for two groups of students who pass/fail CS2.

TABLE I
RANKING OF COURSES BY DIFFERENT SELECTION ALGORITHMS

Attribute	Chi-Squared	Gain Ratio	Info Gain	Relief-F	Correlation	Average rank
CS1	1	1	1	1	1	1
Phys1	2	2	2	2	3	2.2
Phys2	3	3	3	6	2	3.4
Calc2	4	4	4	3	5	4
AdmitGPA	5	5	5	4	4	4.6
Comp1	6	6	6	5	7	6
Comp2	7	7	7	7	8	7.2
Calc1	8	8	8	8	6	7.6

- From the histograms, ‘A+’ or ‘A’ (grade point of 4.0) seems to be the most frequent grade for all of the courses. This makes sense as these student were permitted into CS2 based on AdmitGPA and grade in CS1.
- Success in CS2 has maximum correlation with grade in CS1 of 0.335 while both CS1 and CS2 have minimum correlation with the two English Composition courses.

2) Average grade points for pass and fail: After forming two groups, those who pass and those who fail in CS2, we calculate the average grade points (along with 95% confidence interval) in previous courses for the two groups, as shown in Fig. 2. We can observe that in all of the courses, except for Comp2, students who passed CS2 had a higher average grade point than students who failed. The non-overlapping confidence interval of pass/fail bars for Phys1, Phys2, AdmitGPA, and CS1 indicates that mean grade points were statistically significantly different for these courses. Welch’s t-test (at 0.05 level of significance) confirmed significant difference in mean grade points for Calc2 in addition to the four courses (Phys1, Phys2, AdmitGPA, and CS1) between the two groups. This signifies that these five grade points are good indicators while grades in Calc1, Comp1, and Comp2 do not inform on student success in CS2.

3) Ranking of importance of course grades: The ranking of importance of each course grade in predicting success in CS2 as determined by different algorithms is given in Table I, sorted according to the average rank. Note that the ranking by correlation is determined directly from the Pearson correlation coefficient values in the last column of Fig. 1. Grade points in CS1 rank the highest and is the single most important factor in predicting student’s success in CS2 (determined unanimously by all algorithms), followed by grades in the two Physics courses.

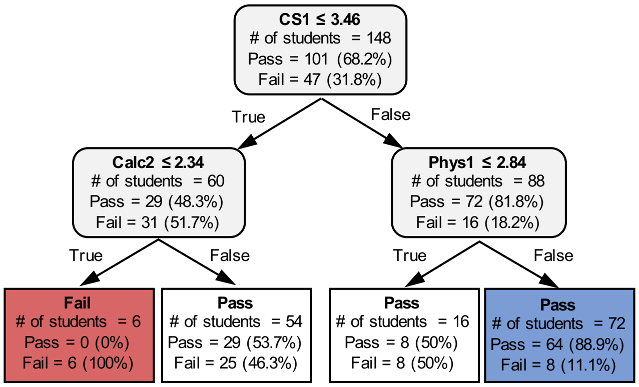

Fig. 3. Decision tree to predict if a student will pass or fail in CS2.

4) Classification with decision tree: Using information gain as the criteria to find the best split, we build a decision tree as shown in Fig. 3. Missing grades in input are replaced with mean values in the corresponding course. The entire data was used for training as we are interested in the rules that the decision tree learns from our data. To avoid overfitting to the training data and create overly complicated rules, we limit the depth of the tree to 2. Each non-leaf node (rectangles with rounded corners) has a condition on top, based on which we split the data. If the condition is true we branch left, else branch right. Leaf node has a label (pass/fail) instead of a condition on top which represents the prediction made by the decision tree. Each node also has information on number of students along with the pass/fail distribution at that particular node. For example, in the root node, CS1 $\leq$ 3.46 is the condition, there are 148 students to start with, out of which 101 (68.2%) of the students passed and 47 (18.2%) failed in CS2. Of the 148 students, 60 go down the True (left) path as they have grade point in CS1 less than or equal to 3.46 and 88 students go down the False (right) path.

The condition at the root selects the most discriminative course grade to predict pass/fail. Grade point in CS1 was chosen by the decision tree as the most important attribute as found earlier. In fact, 31.8% of students fail in CS2 and this figure goes up to 51.7% for students who did not get an ‘A’ in CS1 (CS1 $\leq$ 3.46). Similarly, pass percentage for students is 68.2% in CS2 which goes up to 81.8% for students who got an ‘A’ in CS1. We can get rules from a decision tree by following the path from the root to a leaf. Two interesting rules can be built from the leftmost red leaf and the rightmost blue leaf.

- RULE #1: *IF CS1* $\leq$ *B+ AND Calc2* $\leq$ *C+ THEN Fail.* All 6 students who satisfied this rule in fact failed.
- RULE #2: *IF CS1* $>$ *B+ AND Phys1* $>$ *B− THEN Pass.* 88.9% of the students (64 out of 72) who satisfied this rule passed CS2.

TABLE II
NUMBER OF PASS/FAIL IN CS2 BY GENDER

Gender	Pass	Fail	Total	Pass percentage
Male	83	39	122	68.03%
Female	18	8	26	69.23%

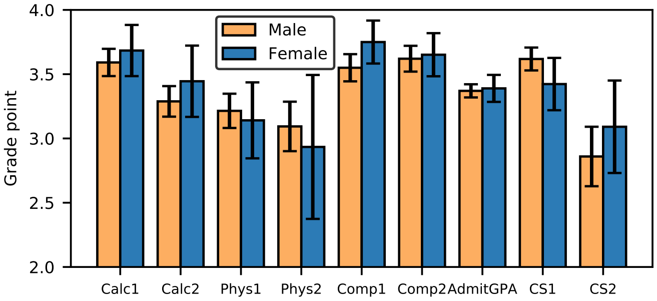

Fig. 4. Average grade points in different courses with 95% confidence interval for male and female students.

B. Effect of Gender

Out of the 148 students in our study, 122 (82.5%) were male and 26 (17.5%) were female, reflecting the lower percentage of women in Computer Science and Engineering programs similar to the national average of 16.3% [8]. In order to study the success rate across gender, we count the number of male and female who pass/fail in CS2. As shown in Table II, pass percentage for both male (68.03%) and female (69.23%) are almost equal to the average pass percentage of 68.2%. We further calculated the average grade points (along with 95% confidence interval) across gender to check if one of the group outperforms the other in any of the courses. Fig. 4 shows that 95% confidence intervals in all of the courses overlap for the two groups. Welch's t-test at 0.05 level of significance showed no significant difference in mean grade points in any of the courses, except for Comp1 in which female students were significantly better than male students.

IV. DISCUSSION

One of the repeating themes from the analysis we performed was that the grade in CS1 is the most important predictor of a student's success in CS2, as found in a previous study [3]. A student with an 'A' in CS1 had an 18.2% change of failing CS2 while if the student had a 'B' in CS1, the student was 2.8 times more likely to fail CS2 with a 51.7% change of failing. Extrapolating this result, we can say that if a student has less than a 'B' in CS1, the student would fail CS2 more than half the time. This justifies the use of grade in CS1 as a requirement to be admitted into our programs.

We also found that grades in Physics were more indicative of success in CS2 than grades in Calculus while the two English Composition courses showed no effect on CS2 success by any measure. When we calculated a modified admission GPA taking only Physics and Calculus grades into account, the correlation with CS2 success increased to 0.316 (from 0.282 with AdmitGPA). Similarly, this modified admission GPA ranked higher in terms of being able to predict student success in CS2, coming second to only CS1 grades. This finding might help the department to update the admission criteria in the future by not including grades in English Composition courses in the admission GPA calculation.

From the decision tree analysis we were able to find a group of six students all of whom failed in CS2. All of these six student had a 'B' in CS1 and a 'C' in Calc2. Three of them got a 'C' in CS2, one had a 'D', one had an 'F', and one had dropped the course. In the future, we could have interventions for such at-risk students (a 'B' or lower in CS1 and a 'C' or lower in Calc2), such as, requiring to attend all classes in CS2 (or be removed from the course for unexcused non-attendance) and/or to participate in a new help session recitation section.

V. RELATED WORK

Our work builds upon previous work that use data-driven techniques to predict student success in both computer programming courses and in computer science degree programs. Azcona and Smeaton [9] used machine learning techniques to identify at-risk students in a CS1 course using logs of students interaction with a virtual learning environment. Estey et al. [10] used an interaction log from a programming practice tool to identify students who struggle in CS1 course. Interaction logs, while useful, require additional data collection and become meaningful only after a few weeks into the course.

We use readily available grades from key previous courses as predictors. Trytten and McGovern [2] also studied grade patterns in introductory courses, including in CS1 and CS2, to see if grades could be used as a predictor for future student success. A somewhat similar study by Kumar [3] analyzed student grades in required computer science courses to predict overall GPA. Both of these studies found that students who graduate from a computer science program had at least a certain minimum grade in a combination of key courses. These past studies primarily used joint distributions and analysis of variance (ANOVA) techniques. Our work complements this past work using modern data mining tools and adds to the body of knowledge of methods for prediction of student success.

VI. SUMMARY AND FUTURE WORK

Using freely available data mining tools, we have shown how it is possible to automatically analyze student course grade data to predict success in CS2. Based on our results, we have suggested evidence-based possible improvements to our program admission criteria and student advising. In the future, we would like to validate our findings on future classes as well as understand the effectiveness of our proposed interventions. We would also like to study the impact of grades in core required computer science courses on student success. Directly replicating findings from existing works (such as [2] and [3]) using the data mining tools used in our study would also be of significant value.

Acknowledgement

We would like to thank the anonymous reviewers for their valuable feedback in improving this paper.

References

[1] E. Frank, M. A. Hall, and I. H. Witten, “The weka workbench,” *Data mining: Practical machine learning tools and techniques*, vol. 4, 2016.

[2] D. A. Trytten and A. McGovern, “Moving from managing enrollment to predicting student success,” in *2017 IEEE Frontiers in Education Conference (FIE)*, Oct 2017, pp. 1–9.

[3] A. N. Kumar, “Predicting student success in computer science – a reproducibility study,” in *2018 IEEE Frontiers in Education Conference (FIE)*, Oct 2018, pp. 1–6.

[4] E. Osmanbegović and M. Suljić, “Data mining approach for predicting student performance,” *Economic Review*, vol. 10, no. 1, pp. 3–12, 2012.

[5] A. M. Shahiri, W. Husain, and N. A. Rashid, “A review on predicting student’s performance using data mining techniques,” *Procedia Computer Science*, vol. 72, pp. 414–422, 2015.

[6] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Wadsworth, 1984.

[7] Scikit-learn: Machine Learning in Python. [Online]. Available: https://scikit-learn.org/stable/

[8] S. Zweben and B. Bizot, “2015 Taulbee Survey: Continued Booming Undergraduate CS Enrollment; Doctoral Degree Production Dips Slightly, Computing Research News,” pp. 1–59, 2016.

[9] D. Azcona and A. F. Smeaton, “Targeting at-risk students using engagement and effort predictors in an introductory computer programming course,” in *Data Driven Approaches in Digital Education*. Springer International Publishing, 2017, pp. 361–366.

[10] A. Estey, H. Keuning, and Y. Coady, “Automatically classifying students in need of support by detecting changes in programming behaviour,” in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE ’17, 2017, pp. 189–194.