APPLICATIONS OF MACHINE LEARNING TO STUDENT GRADE PREDICTION IN QUANTITATIVE BUSINESS COURSES

Timothy Anderson, Stanford University
Randy Anderson, California State University, Fresno

ABSTRACT

With ever growing pressure to increase student performance, as well as the need to support first generation students and those from underrepresented backgrounds, never before has there been a greater need to quickly identify students who are on track to perform poorly in a class. Traditionally, instructors have relied mainly on intuition and rough averaging of exam scores to predict what grade a student who is on track to receive. With the advent of machine learning and data mining, we have the unprecedented ability to leverage historical student performance data to make statistically accurate and justified predictions beyond an instructor's intuition. Here, we demonstrate the efficacy of three machine learning algorithms—Naïve Bayes estimation, K-nearest neighbors, and support vector machine—for use in predicting students' grades, and compare the results to a "simple average" approach of predicting student grades as a proxy for an instructor's prediction. We show that a support vector machine outperforms all other predictors, including the simple average. However, the error of the simple average predictor is not significantly greater than with the support vector machine, and with the increased computational cost of the support vector machine over the simple average predictor, it may be more economical and convenient to instead rely on instructor predictions or a simple average predictor than a more complex algorithm. Overall, while we show here that machine learning is promising for applications in predicting student performance outcomes, there is still much work to be done towards applying this technology on a broad scale.

INTRODUCTION

At some point, every college instructor has inevitably been asked by a student midway through the semester: "What grade am I going to get in your class?" The most naïve approach is to simply calculate the student's current grade, and assume this as their final grade. However, this approach fails to take into account factors such as variations in difficulty between exams, performance trends over the course of the semester, and similarities and differences between past and present students. Every instructor who has taught the same class multiple times has accumulated significant amounts of past grading data. Here, we ask:" Is it possible to meaningfully predict students' grades using only past grading data?"

Similar studies have examined this question. Majeed (2016) examined the effect of taking both the instructor's grading history as well as past student performance into account when predicting grades, and was able to achieve 95% classification accuracy by using a large amount of

data and features to perform this classification. Lakshmi (2013) employed a genetic algorithm to analyze the root causes of student performance and behavior.

To predict students' grades across our data set, we employ several machine learning algorithms. We compare the results of the machine learning algorithms to the "simple average" approach most often employed by professors and students to make predictions. What follows is a brief overview of machine learning methods, results of using machining learning methods to predict students grades in comparison to a baseline "simple average" prediction, and a discussion of pedagogical applications of this approach to grade prediction.

MACHINE LEARNING OVERVIEW

A. What is machine learning?

Machine learning, in its simplest definition, is using data and statistics to make meaningful predictions about other data. Much like learning in humans, a machine learning algorithm will look for patterns in existing data or train itself to correlate certain inputs with certain outputs. There are two main families of machine learning algorithms: unsupervised and supervised learning. Unsupervised learning seeks to find structure in existing data. Most often, this is focused on looking for clusters of data points within a data set. An example would be looking at gene expression data to see if patients tend to cluster into different groups based on genetic data. Tair (2012) showed the usefulness of using unsupervised learning techniques to analyze student data. In this study, they used techniques such as singular value decomposition to discover trends in the data.

However, while unsupervised techniques can give insight into the structure of data, it cannot in general make exactly predictions from data. To this end, we turn to supervised learning. Supervised learning seeks to correlate inputs with outputs to train a model that predicts the output for a given input. The simplest example of supervised learning is linear regression: taking a set of input and output data points and fitting a best-fit line. However, supervised learning also describes a wide range of models, ranging from simple regression models to something as complex as convolutional neural networks, a model often used in computer vision or natural language processing.

For a classification problem in machine learning, we have n different classes of data in our *training set*, or the dataset used to optimize our statistical model, and we would like to find the class of data in our *test set*, or set of data not used to train the model to mimic empirically applying the model. Grade prediction is such a classification type problem: we have exam scores from throughout the semester for students, and we would like to predict their final grade in the course. To perform this task, we employ several different supervised machine learning algorithms.

B. Supervised learning methods

Agrawal (2015) explored the use of on a Naïve Bayes classifier for student performance prediction. In this study, we employed two additional supervised learning methods to make predictions about student grades: k-nearest neighbors and support vector machine.

i. Naïve Bayes

Naïve Bayes estimation is based on Bayes rule:

$$p(C_k|\boldsymbol{x}) = \frac{p(C_k)p(\boldsymbol{x}|C_k)}{p(\boldsymbol{x})}$$

where C_k represents a class of data (e.g. a student's grade) and $\boldsymbol{x}$ is our feature vector, the numerical values we use to characterize a data point (James 2013). Naïve Bayes is "naïve" in the sense that we assume independence between the constructs in our feature vector. By making this assumption, we can rewrite Bayes rule as:

$$p(C_k|x_1, x_2, \ldots, x_n) \propto p(C_k) \prod_{i=1}^{n} p(x_i|C_i)$$

From here, we can pick our prediction $\hat{y} = C_k$ i.e. the class which maximizes the probability. In mathematical terms, this is:

$$\hat{y} = \underset{C_{k=1,\ldots,n}}{\arg\max}\, p(C_k) \prod_{i=1}^{n} p(x_i|C_i)$$

ii. *K*-nearest neighbors

K-nearest neighbors classifiers classify a data point as belonging to the class it is geometrically closest to in our training dataset (James 2013). The different classes of data are shown in color, and our data point of interest is in green. In *K*-nearest neighbor classification, we first calculate the distance between the point we wish to classify and the points in our training dataset. From here, we look at the *K* closest points in the training set, and pick our point as belonging to the most common class in the *K* nearest points.

An example of *K*-nearest neighbors classification is shown in Figure 1 below. The point of interest is in green, and we wish to classify it as belonging to the red or blue class. After calculating the distance between the green point and the other points in our data set, we can do a majority vote with the *K*-nearest points. In this case, if we use *K* = 1, 2, or 3, then we will have Class = Red. However, if we use *K* = 5, we have Class = Blue. This demonstrates that the *K*-nearest neighbors algorithms will not deliver the same classification depending on number of nearest neighbors used.

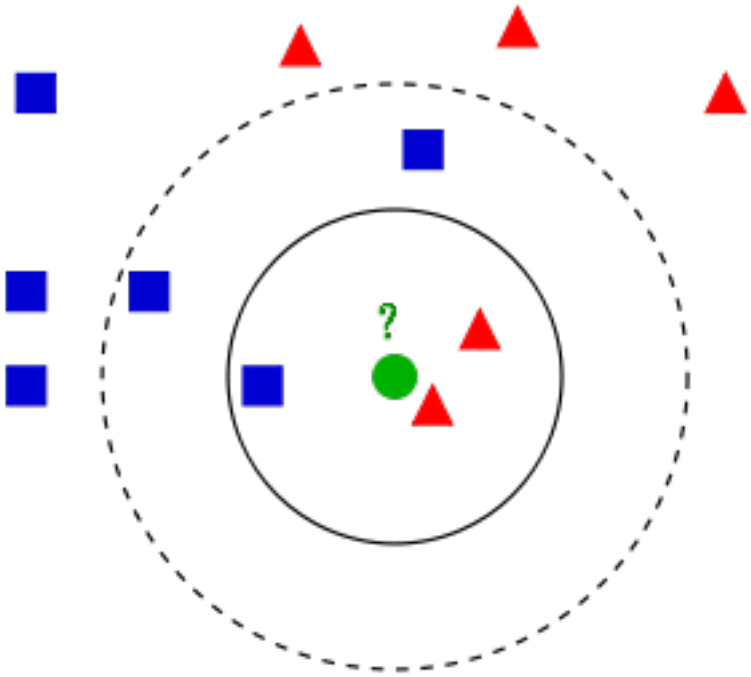

Figure 1: Illustration of K-nearest neighbors. Here, we want to determine the class of the green point. If we use K = 1, then it will be classified as red. However, were we to use K = 5, the majority nearest neighbors would be blue, so the green dot would be classified as belonging to the blue class. (Source: Wikimedia Commons)

iii. Support Vector Machine

The final type of classifier we employ in this study is a support vector machine (SVM). A SVM seeks to find a maximal separating hyperplane between two classes of data (Hastie 2009). In two dimensions, this is equivalent to finding a line to separate or minimize the overlap between two types of data. This is shown in Figure 2 below. Here we have two classes of data: filled and empty dots. The red line shown is the maximal separating hyperplane since it maximizes the distance between each data point and the line. SVMs are extremely effective at separating data from different classes and have been a staple of machine learning algorithms since the advent of their modern form in the 1990s (Cortes 1995).

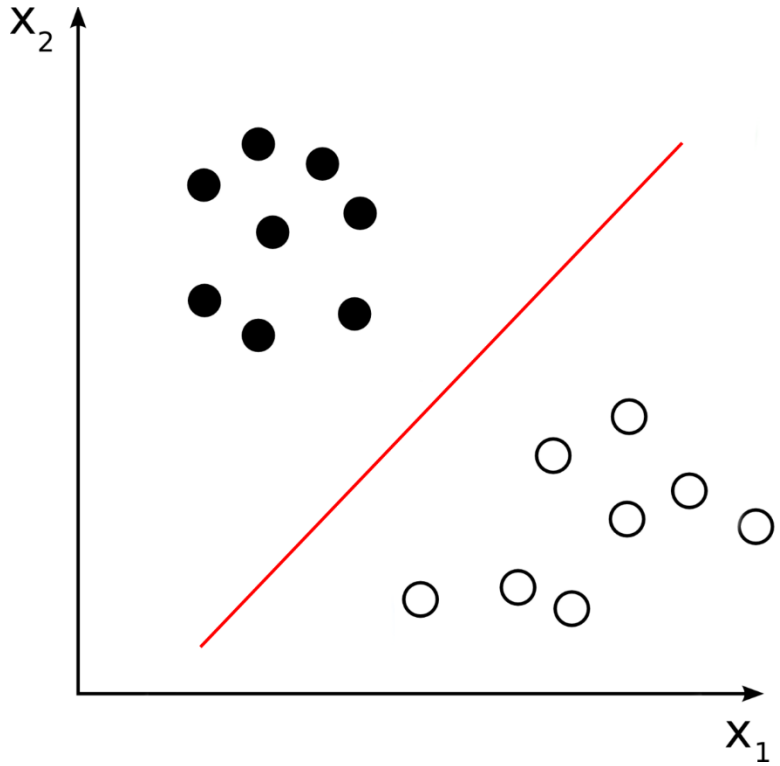

Figure 2: Illustration of maximal separating hyperplane. In an SVM, we seek the line which can separate two classes of data the most strongly i.e. has the maximum distance from the line to the nearest points in each class. (Source: Wikimedia Commons)

In cases where we have multiple classes of data—e.g. in the case of students' grades—we can use a series of "one on one" SVM multi-class classifier. With this approach, we create SVM classifiers between all classes and each other (for a grading scale from A to F, this would employ 10 total SVMs to classify the five different grades). From here, we classify a given data point with every SVM in our set of one-on-one SVMs and pick the majority class.

GRADE PREDICTION RESULTS

To perform the classification, we use historical grade data from 18 semesters totaling 683 students (N=683). The grade data were compiled from statistics courses offered at the Craig School of Business at California State University, Fresno from 2006 to 2015. This dataset was chosen such that the structure of the class had not changed significantly over this time period, allowing for this dataset to be compiled into a single, large dataset. In this class, students were evaluated based on four unit examinations and one final exam. The overall distribution of grades for the entire dataset is summarized in Figure 3 below.

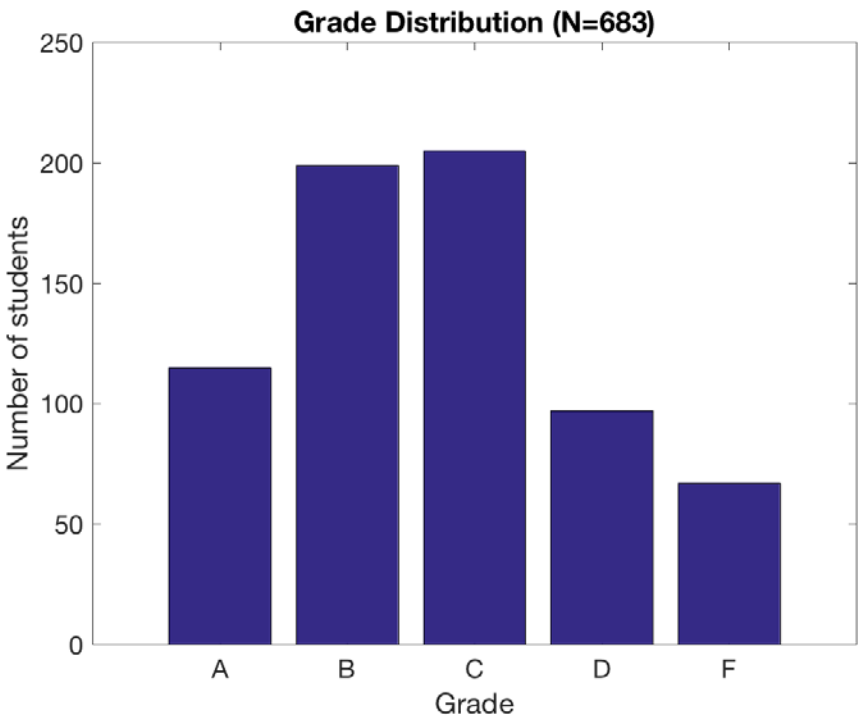

Figure 3: Grade distribution for dataset. The distribution exhibits very little skew, with most students receiving B's and C's in the course.

To perform the grade prediction, we use the three classifiers described in the previous section: Naïve Bayes, K-nearest neighbors, and SVM. We can think of each classifier as looking at a given student whose grades we would like to predict, and seeing which class of grades they are most likely to be a part of. In essence, we are predicting a given student's grade based on how students in the past have performed.

The errors rates of the classifiers are the average error rates found by using 10-fold cross validation. "*k*-fold" cross validation is a process whereby the dataset is partitioned into a training set (~90% of the dataset for the study presented here) and a testing set of the remaining data (10% of the overall dataset), and the error rate calculated for *k* different training and testing sets drawn from the same data. For comparison, we also predicted students' grades using a simple average of their exam scores. The simple average classifier averages a student's performance across exam scores and maps this average to the corresponding final grade (e.g. a student with a 91% average on the first three exams will have a predicted final grade of "A").

Along with comparing different predictors, we have also compared the prediction accuracy using different amounts of information. It is natural to want to predict a student's grade throughout the semester, so we have tested all of the classifiers using one to four exams to mimic predicting students' grades with limited information. The error when performed on the training and testing datasets for each classifier for each progressively more exams is shown in the plots below (Figures4 and 5).

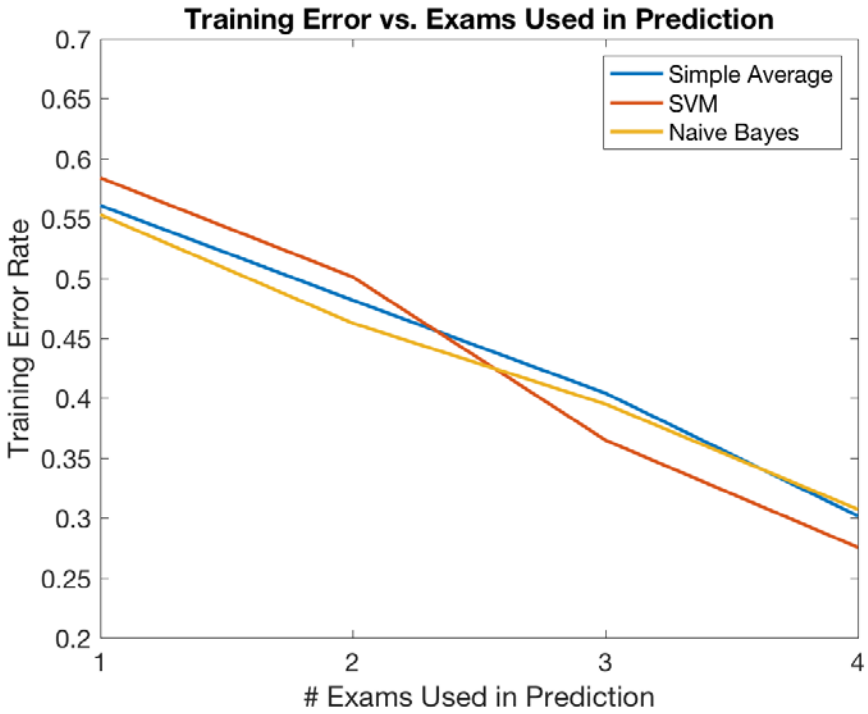

Figure 4: Summary of training error for our predictors. The SVM and Naïve Bayes classifiers slightly outperform our baseline simple average classifier. Notes: K-nearest neighbors is excluded since it is not practical to find a training error for this classifier.

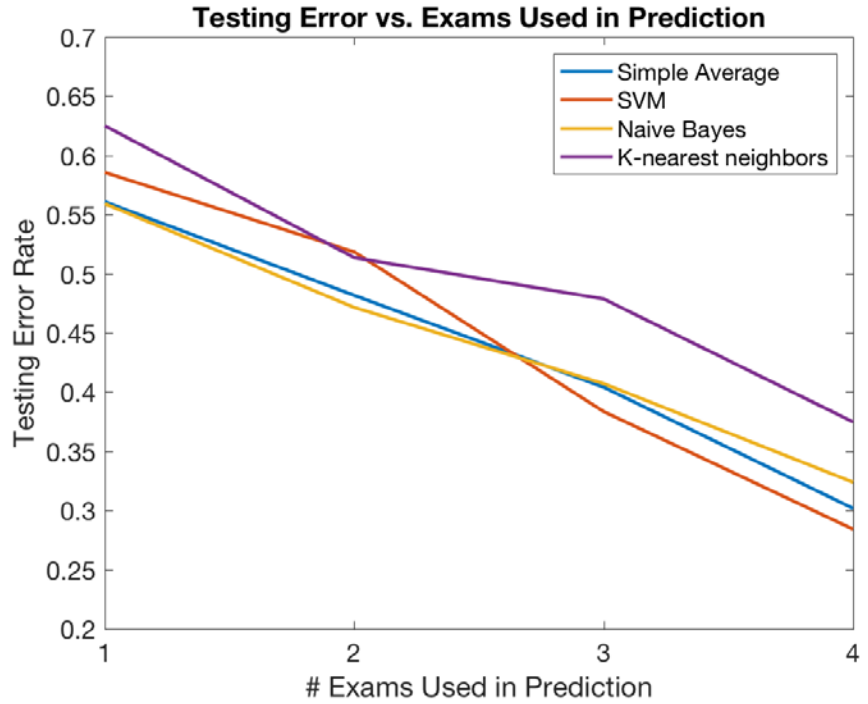

Figure 5: Summary of testing error for our predictors. The simple average has comparable performance or outperforms the classifiers besides SVM.

DISCUSSION AND PEDEGODICAL APPLICATIONS

The results show that from the perspective of obtaining the lowest error rate, a SVM is the best classifier for data of this kind. Furthermore, it is the only classifier that consistently outperforms a "simple average" approach to predicting a student's grade. This could be due to several effects. First, the SVM classifier actually creates 10 different one-on-one classifiers and picks the class through a majority vote approach. By building multiple classifiers, we are able to achieve a more optimal classifier for each class of data, compared to Naïve Bayes or k-nearest neighbors where we use one classifier for all data classes. Secondly, due to the large variability of student performance within each class of data—that is, two students who receive a "B" could have vastly different exam grades—the *K*-nearest neighbor classifier and Naïve Bayes will have diminished performance. For *K*-nearest neighbors, we are assuming if a student performs closely with other students on each exam, that student will also perform similarly on the final exam as the nearest other students. Thus, this student will receive the same final grade in the class. Similarly for Naïve Bayes, we are relying on each class of students to have strong separation based on their exam scores. In the real classroom, these assumptions are simply not true: students perform better or worse than expected on exams for a wide variety of reasons, some beyond the classroom setting. Other than variability in testing performance, student grades exist within a spectrum: while a student who earns a 79% and another who earns an 80% have very similar performance in the class—and thus are viewed the same by an SVM or *K*-nearest neighbors classifier—the former would receive a "C" and the later a "B," showing that very little numerical separation in performance can and does lead to disparate grades.

The other main reason for the diminished performance of the classifiers in comparison to the simple average classifier is issues with the data set. While the pedagogical approach in this statistics course used to test these classifiers had the same structure over several years, exam difficulty did inevitably vary between years. These classifiers rely on greater consistency between exams in different years. For example, we are assuming that the grade distribution in 2007 is similar to that in 2014, and, furthermore, that the distribution of student performance on individual exams does not differ significantly between years. Because these assumptions only hold to a limited extent, the performance of our classifiers ultimately suffers, and, in general, it becomes nearly impossible to predict student performance based solely on grade data if there a larger than expect variability in data drawn from different years.

The primary application of this work is empowering students and instructors to actively predict a student's final grade in a class based on their performance before the final exam. For instructors, this is useful for identifying students at risk of performing poorly in the class so they can recommend or require additional tutoring. Currently in higher education, there is mounting pressure to increase student performance. As shown in Pandey (2011), it is very easy to identify under-performers in a dataset using a Naïve Bayes approach to prediction. With many schools launching special programs to increase the proportion of first generation or students from underrepresented backgrounds, the ability to actively predict students' grades throughout the semester and identify those who are struggling would be essential in the effort to optimally allocate resources for additional tutoring for at-risk students. On the students' side, being able to predict

their final grade based on exam scores would allow them to better allot their study time between courses, as well as make informed decisions about enrolling or remaining in a given course. By employing machine learning, we can provide more statistically accurate and justified information to students and teachers, and in turn enhance the classroom experience on both sides.

However, while this technology does have many advantages for students and instructors, we must keep in mind the practical constraints at hand for implementing such a workflow. For a very large set of data—say, across several thousand students over many years—calculating a SVM or Naïve Bayes predictor can be somewhat computationally expensive for a desktop computer or mobile device. Furthermore, such datasets may not be available due to significant changes in a course's structure over time or may be difficult to obtain due to the format of historical grade data. Lastly, as shown by the results presented here, a simple average classifier performs nearly as well as significantly more complicated algorithms at a fraction of the computational cost. In essence, bigger is not always better when it comes to classifying data of this type. As a corollary to this, because we intuitively use a simple average classifier when predicting students' grades, a more complex classifier may not perform significantly better. While our results validate the simple average approach, it also shows that it is a good proxy for much more complex methods: indeed, the accuracy gained from using a method such as a SVM may not be worth the additional computational cost. Overall, based on our results, machine learning does show promise for use in grade prediction, it will be necessary to increase the accuracy significantly beyond a simple average classifier to make it an attractive technology for use in the classroom.

CONCLUSION

Machine learning stands to revolutionize education in the near future: by employing machine learning to tasks such as designing curriculum, customizing pedagogy, and tracking and predicting student performance, we will be able to enable a new era of engagement with students. This study puts forth an example of using machine learning to predict students' grades in a quantitative business course. The results show that while machine learning is promising for predicting students' grades, the gains from employing supervised learning algorithms may not be worth the additional computational cost. Overall, while machine learning is a very powerful technology—especially when applied to education—there is still much work to be done in applying it in the classroom.

REFERENCES

Agrawal H and Mavani H (2015). "Student Performance Prediction using Machine Learning". International Journal of Engineering Research and Technology. 4 (03):111–113.

Cortes C, Vapnik V (1995). "Support-vector networks". Machine Learning. 20 (3):273–297.

Hastie T, Tibshirani R, and Friedman J (2009). "The Elements of Statistical Learning: Data Mining, Inference, and Prediction". New York: Springer.

James G, Witten D, Hastie T, Tibshirani R (2013). "An Introduction to Statistical Learning: with Applications in R". New York, NY: Springer.

Lakshmi TM, Martin A, and Venkatesan VP (2013). "An Analysis of Students Performance Using Genetic Algorithm." Journal of Computer Sciences and Applications. 1 (4):75–79.

Majeed EA, Junejo KN (2016). “Grade Prediction Using Supervised Machine Learning Techniques”. e-Proceeding of the 4th Global Summit on Education 2016. 222–234.

Pandey UK and Pal S (2011). “Data Mining: A prediction of performer or underperformer using classification”. International Journal of Computer Science and Information Technology. 2 (2):686–690.

Shahiri AM, Husain W, and Rashid NA (2015). “A Review on Predicting Student’s Performance Using Data Mining Techniques.” Procedia Computer Science. 72:414–422.

Tair MMA and El-Halees AM (2012). “Mining Educational Data to Improve Students’ Performance: A Case Study”. International Journal of Information and Communication Technology Research. 2 (2).