Feature extraction for classifying students based on their academic performance

Agoritsa Polyzou
Computer Science & Engineering Department
University of Minnesota
Minneapolis, MN 55454, USA
polyz001@umn.edu

George Karypis
Computer Science & Engineering Department
University of Minnesota
Minneapolis, MN 55454, USA
karypis@umn.edu

ABSTRACT

Developing tools to support students and learning in a traditional or online setting is a significant task in today's educational environment. The initial steps towards enabling such technologies using machine learning techniques focused on predicting the student's performance in terms of the achieved grades. The disadvantage of these approaches is that they do not perform as well in predicting poor-performing students. The objective of our work is two-fold. First, in order to overcome this limitation, we explore if poorly performing students can be more accurately predicted by formulating the problem as binary classification. Second, in order to gain insights as to which are the factors that can lead to poor performance, we engineered a number of human-interpretable features that quantify these factors. These features were derived from the students' grades from the University of Minnesota, an undergraduate public institution. Based on these features, we perform a study to identify different student groups of interest, while at the same time, identify their importance.

Keywords

academic student success, classification, feature importance

1. INTRODUCTION

Higher educational institutions constantly try to improve the retention and success of their enrolled students. According to the US National Center for Education Statistics [8], 60% of undergraduate students on four-year degrees will not graduate at the same institution where they started within the first six years. At the same time, 30% of college freshmen drop out after their first year of college. As a result, colleges look for ways to serve students more efficiently and effectively. This is where data mining is introduced to provide some solutions to these problems. Educational data mining and learning analytics have been developed to provide tools for supporting the learning process, like monitor and measure student progress, but also, predict success or guide intervention strategies.

Most of the existing approaches focus on identifying students at risk who could benefit from further assistance in order to successfully complete a course or activity. A fundamental task in this process is to actually predict the student's performance in terms of grades. While reasonable prediction accuracy has been achieved [14, 10], there is a significant weakness of the models proposed to identify the poor-performing students [18]. Usually, these models tend to be over-optimistic for the performance of students, as the majority of the students do well, or have satisfactory enough performance.

In this paper, we investigate the problem of predicting the performance of a student in the end of the semester before he/she actually takes the course. In order to focus on the poor-performing students, who are the ones that need these systems the most, the prediction problem is formulated as a classification task, where two groups of students are formed according to their course performance. We essentially identify two complementary groups of students, the ones that are likely to successfully complete a course or activity, and the ones that seem to struggle. After identifying the latter group, we can provide additional resources and support to enhance their likelihood of success.

However, "success" and "failure" can be relative or not. For example, a B− grade might be considered a bad grade for an excellent student, while being a good grade for a very weak student. We investigated different ways to define groups of students taking a course: failing students, students dropping the class, students performing worse than expected and students performing worse than expected, while taking into consideration the difficulty of a course.

In order to gain more insight into the learning process and its most important characteristics, we have created features that capture possible factors that influence the grades at the end of the semester. Using these features, we present a comprehensive study to answer the following questions: which features are good indicators of a student's performance? which features are the most important? The findings are interesting, as different features are the most important for different classification tasks.

The rest of the paper is organized as follows. Section 2 reviews the work in the area of predicting student performance

in the end of the semester. In Section 3, there is an overview of the data that we used. Section 4 describes the features extracted, and Section 5 the classification tasks and methods tested. In Section 6, there is a detailed discussion of the experimental evaluation of the different methods tested, as well as the feature importance study. Section 7 contains the conclusions of the study.

2. RELATED WORK

As we are interested in estimating next-term student performance, we will review the related work in this area of research. The binary classification has been used in various educational problems, like predicting if a student will drop out from high school [6] or to predict if a student will pass a module in a distance learning setting [7]. Multi-label classification has been applied to provide a qualitative measure of students' performance. In [17], decision tree and naive Bayes classifiers are used with data from a survey. Attributes collected by a learning management system have been employed to estimate the outcome as Fail, Pass, Good and Excellent [16], or to classify students [12]. Some approaches [11, 9] test different ways to label the student performance, with two (pass or fail) or more labels. The majority of the aforementioned approaches are small-scale studies, that are applied to a limited number of courses.

In recent years, influenced by advances in the recommender systems, big data approaches have been also utilized in the area of learning analytics. Initially, the term "next-term grade prediction" was introduced by Sweeney et al. [18] in the context of higher education, and it refers to the problem of predicting the grades for each student in the courses that he/she will take during the next semester. Models based on SVD and factorization machines (FM) were tested. In another approach [15], the previous performance of students controls the grade estimation in two different ways while building latent models. In [19], some additional state-of-the-art methods were used, as well as, a hybrid of FM and random forests (RF). The data used are the historical grades and additional content features, representing student, course and instructor characteristics. At the same setting, [14] and [10] developed course-specific methods to perform next-term grade prediction based on linear regression and matrix factorization.

All these methods assign a specific numerical grade to each student's attempt to take a course. A limitation identified in these approaches was that the developed models perform poorly for failing students. In [5], failing students have been completely removed from the dataset. As this is the subpopulation of students that needs additional support the most, it is very important for a model to be able to accurately identify these students at risk.

This work is a more general study of the factors that influence the student performance, in a very large scale. The only observed data that we have available are the students' grades at the end of the semester. In our approach, we formulated this problem as a binary classification task, in order to detect the different group of students. In other words, we keep the classification methodology, but apply it on the context of big data.

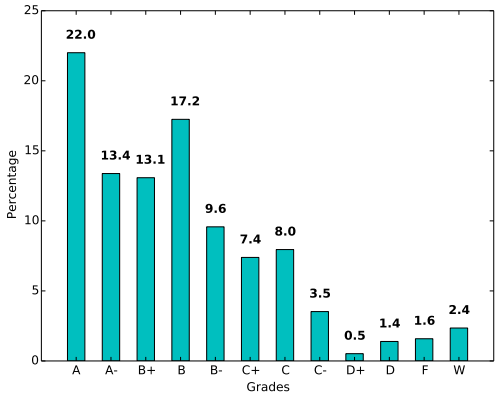

Figure 1: Percentage of each letter grade with respect to the total grades.

3. DATASET

First, we will clarify the use of some terms in the current context. An *instance* refers to the performance of a student, s, in a course, c, at the end of the semester. All the courses that a student took in past semesters, before taking course c, are the *prior courses*, denoted by $C_{s,all_{prior}}$. The set of courses for a single semester x is denoted as $C_{s,x}$. Additionally, for a course c there might exist a stated set of courses that are required for a student to take before attempting c. We refer to this set as the *prerequisite courses*. Every course x worths a specified number of credits, cr_x.

An undergraduate student enrolled to a college or university has to take some courses each semester, and receive a satisfactory grade in order to successfully complete them. Depending on the student's degree program, these courses might be required, electives, or simply courses that the student takes for his/her own advancement, intellectual curiosity, or enjoyment. If a student withdraws from a course after the first two weeks of classes, it is denoted by the letter 'W' in the student's transcript.

The original dataset was obtained from the University of Minnesota and it spans over 13 years. We removed any instances with a letter grade not in the A–F grading scale (A, A−, B+, B, B−, C+, C, C−, D+, D, F). Statistics about the grades in the dataset are shown in Fig. 1. In our dataset, the letter grade A is the most common. We extract features for the instances occurring during the last 10 fall and spring semesters. Given a semester, we utilize all the students that had taken the course before, and for each student taking a course, we extract a set of features. Additionally, we generate features for the instances awarded with the letter W, but we do not utilize them in any other way during the feature extraction process. These will be used only when trying to predict the students that drop-out from a course.

4. EXTRACTED FEATURES

Having as input the historical grading data, we derived different features to capture possible factors for a student's poor performance. The features can be separated into three distinct categories: the student-specific (independent from course c), course-specific features (independent from student

s) and student- and course-specific features (they are a function of both s and c). All extracted features are described in Tables 1, 2, where related features are grouped together into eight different subcategories. The keywords on bold are used to indicate the corresponding group of features later. Note that for each $\{s, t, c\}$, where student s took course c in semester t, we generate a different set of features. Every set of features characterize a student's attempt to take course c at the specific point of his/her studies.

These features are either numerical, categorical or indicator variables. For indicator features, we use the values of 0 or 1. The categorical features are encoded via a numerical value. For example, the feature about the current semester is categorical, and the values {fall, spring, summer} are transformed to {0,1,2}, respectively.

5. CLASSIFICATION PROBLEMS

5.1 Classification tasks

Our motivation was to identify groups of students that need further assistance and guidance in order to successfully complete a course. These students could benefit from informed interventions. We consider this to be a binary classification problem, where these students form one of the classes and the remaining students form the other class.

We consider different ways of measuring when a student does not do well in a course to deal with the performance measurement challenges we mentioned earlier. Unsatisfactory performance can occur when the earned grade represents a performance that is bellow the student's potential. We considered the following four ways for labelling, resulting to these absolute and relative classification tasks:

1. Failing student performance, i.e., letter grades D and F (denoted as the **Fgr** task).

2. The letter grade W (denoted as the **Wgr** task). This represents the instances when the student dropped the course. This behavior is worrisome as it shows that either the student was not interested in the course anymore or he/she expects to perform poorly.

3. Student performance that is worse than expected, i.e., the grade achieved is more than two letter grades lower than the student's GPA (denoted as the **RelF** task).

4. Student performance that is worse than expected while taking into consideration the difficulty of the course (denoted as the **RelCF** task). The difficulty of a course is expressed by the average grade achieved by the students that took the course in prior offerings. A positive instance is when the grade achieved is more than two letter grades lower than the average of the student's GPA and the course's prior average grade.

Statistics for the different classification tasks can be found at Table 3.

As discussed at the related work section, it is easier to predict the successful students. In order to have a better understanding of the relative difficulty of this task compared with the four tasks mentioned above, we also examined the task of predicting the students that completed a course with the grade A (denoted as the **Agr** task).

5.2 Methods compared

In order to support students that need help to successfully complete a course, we will use classification techniques to identify them from the rest of the students. The instances of interest will be labeled as 1, and the rest as 0. The problem can be described as follows. We are given a set of training examples that are in the form $(\mathbf{x}, y)$ and we want to learn their structure. We assume that there is some unknown function $y = f(\mathbf{x})$, that corresponds the feature vector $\mathbf{x}$ to a value y. In our case, $y = \{0, 1\}$. A classifier is an hypothesis about the true function f. Given unseen values of $\mathbf{x}$, it predicts the corresponding y values.

We tested the following classifiers [4], using scikit-learn library in Python [13]: Decision Tree (DT) [2] and Linear Support Vector Machine (SVM) [3] as base classifiers, and Random Forest (RF) [1] and Gradient Boosting (GB) [4] as ensemble classifiers.

While using DT, the classification process is modeled as a series of hierarchical decisions on the features, forming a tree-like structure. In other words, we ask a series of questions about the features of an instance, and based on the answer, we may ask more questions, until we reach to a conclusion about the class label of that instance. The goal is to get a split that allow us to make a confident prediction. Consider the m-dimensional space that is defined by the feature vectors $\mathbf{x}$, of length m. There, every training instance corresponds to a single point. A Linear SVM looks for a decision boundary between two classes, a hyperplane that bisects the data with the largest possible margin between the two different classes. The margin on each side of the hyperplane is the area with no data points in it.

Ensemble methods try to increase the prediction accuracy by combining the results from multiple base classifiers. RF is a class of ensemble methods that uses decision trees as week learners. Randomness has been explicitly inserted in the model building process, as every splitting criterion considers only a subset of features, randomly selected from the feature vector of $\mathbf{x}$, to select the best split. Once we build all the trees, the majority class is reported. In boosting, a weight is associated with each training instance. Using the same algorithm, classifiers are training on a weighted training set to focus on hard-to-classify instances. At the end of each iteration, the weights of instances with high misclassification error are relatively increased for future iterations. In GB for binary classification, a single regression tree is built, where in each splitting criterion, only a subset of the features is considered. Once the tree is built, then, the corresponding weight of the classifier in the current iteration is estimated.

6. EXPERIMENTS

6.1 Experimental design

The models constructed are global, i.e., a single model predicts the performance of all students over all the courses. All features are extracted for any instance of a student taking a course. As randomization takes part in the models while sampling and/or initialization, we run the same model with

Table 1: Feature groups describing the target student s in the target semester t.

(1) Student's status in terms of grades. (**grades**)	• Average grade of s in prior courses C_{s,all_prior}. $\sum_j g_{s,j}/\lvert C_{s,all\_prior}\rvert$, for j in C_{s,all_prior}. • GPA of s, i.e., weighted average of the grades in prior courses w.r.t. the credits worth. $\sum_j g_{s,j} cr_j / \sum_j cr_j$, for j in C_{s,all_prior}. cr_j is the number of credits of course j. • GPA of s over the prior courses that belong in his/her major. • GPA of s over the prior courses that do not belong in his/her major. • GPA of s over the courses taken the previous semester, i.e. at the semester (t-1). • GPA of s over prior courses taken the past two semesters i.e. at semesters (t-1) and (t-2). • GPA of s over prior courses taken on fall, spring and summer semesters. Essentially, here there are 3 features, one for each semester type. • Average grade of courses that s took with the same corresponding credit. There are 6 different features, each corresponding to prior courses that worth 1,2,3,4,5 or 6 credits. • GPA of courses that s took at the same course level. There are 6 levels (1xxx, 2xxx, 3xxx, 4xxx, 5xxx, or 8xxx). Higher level courses are more advanced.
(2) Other info indicating a student's status. (**status**)	• The number of prior courses, $\lvert C_{s,all\_prior}\rvert$. • Student's major. Included majors: Aerospace Engineering, Biomedical Engineering, Chemical Engineering, Chemistry, Civil Engineering, Computer Science, Electrical Engineering, Materials Science, Mathematics, Mechanical Engineering, Physics, and Statistics. • The total credits that s has earned in prior courses. $\sum_j cr_j$, for j in C_{s,all_prior}. • Indicator whether target semester t is a fall, spring, or summer semester. • Indicator whether the student has ever registered for the summer semester. This is an indicator of the past behavior of the student. • The number of semesters that the student is active, $\text{nterms\_active}_{s,t}$. • The number of years student s is in the program. • The number of transferred credits. It is quite common for students to transfer some credits from other institutions, or from qualified courses they took at high school.
(3) Student's course load. (**load**)	• Average credits s earned in prior courses per semester. $\sum_j cr_j/\text{nterms\_active}_{s,t}$, for j in C_{s,all_prior}. • The number of credits s earned in the past semester. $\sum_j cr_j$, for j in $C_{s,t-1}$. • The number of credits earned in the current semester. $\sum_j cr_j$, for j in $C_{s,t}$. • The number of courses taken the current semester. $\lvert C_{s,t}\rvert$. • Ratio of s's course load in the current semester to his/her average course load over the past semesters. This is a way to compare the usual load of the student with the load for the target semester. $(\sum_i cr_i/(\sum_j cr_j/\text{nterms\_active}_{s,t})$, for i in $C_{s,t}$ and j in C_{s,all_prior}.

The set C_{s,all_prior} represents the courses that the student took all the prior semester, before the target semester t. For any semester x, $C_{s,x}$ represents the set of courses that student s took on semester x.

5 different seeds and average out the performance achieved. We used cross validation for classifier evaluation. The data are partitioned into 5 disjoint subsets. For each fold, test on one partition and use the remaining ones for training. The average of the evaluation metrics across the 5 folds will be the values reported.

Metrics. Precision is the ratio of true positives to all predicted positives. Recall is the ratio of true positives to all actual positives. Precision is intuitively the ability of the classifier not to label as positive a sample that is negative, while recall is the ability to find all the positive samples. F_1 score is a measure of accuracy, calculated as:

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (1)$$

Area under the receiver operating characteristic (ROC) curve, AUC, is also reported to understand the performance of a classifier w.r.t. all the thresholds. ROC curve plots the true positive rate against the false positive rate, at various thresholds. AUC corresponds to the probability that the classifier will rank a random positive instance higher than a negative one.

Estimating positive threshold. Instead of assigning a label to a test instance, we can assign a prediction score in the range of [0,1] that will be the probability of the input samples to belong to the positive class. In this way, we will be able to compute metrics like AUC. To estimate a threshold of the prediction score above which the object is assigned to the positive class, we follow these steps: 1. Sort the prediction scores in non-increasing order. 2. For each point L in this sorted sequence, compute the F_1 score, using Eq. 1, by assuming that any instances that have a prediction score that is greater than that of the Lth instance is classified as positive and everything else is classified as negative. 3. The F_1 score is the maximum F_1 value obtained above.

Table 2: Feature groups describing the student s in term t and course c.

(4) Course's difficulty and popularity. (**c-diff**)	• Relative course load when s took c w.r.t. the average credits of past students at the semester they had taken c. For each past student, compute the number of credits earned on that semester. Then, compute the fraction of $\sum_j cr_j$, for j in $C_{s,t}$, divided by the average credits earned from past students on the same semester that they took course c. Values greater than 1 indicate heavier load than other students. • Average grade earned by past students. • Average grade in c of past students within the same major as the s. Now, filter the students in order to keep only the students that are in the same department as s. • Average grade in c of students belonging to c's major or not. This describes two features, by separating the past students to the ones that are in the same major as the department of c, and the ones that are out-of-the-department.
(5) Performance / Familiarity with the course's background and department. (**c-backgr**)	• Fraction of students in the same major as s that have taken the c. This feature measures how popular is course c across the students on the department of student s. • Fraction of students from s's major that took c, shows how common is c in s's major. • Number of courses that s took and belong to c's department. Absolute measurement of how familiar is s with the department of the course c. • Ratio of courses that s took and belong to c's department. Relative measure of how familiar is s with the department of the course c. • Ratio of credits that s took and belong to c's department. Relative measurement of how familiar is s with the department of the course c, in terms of credits. • Ratio of credits that s took and belong to c's department and the average credits that past students took and belonged to c's department. This is a relative measurement of how familiar is s with the department of the course c, in comparison with past students. • GPA over the courses that s took and belong to c's department. This feature is a quantitative measure of student's performance in the c's department.
(6) Information about the prerequisites. (**prerequ**)	• GPA of the prerequisite and non-prerequisite courses that s has taken. Two features that show the performance of the student in prerequisite and other courses. • Number of the prerequisite courses taken by s, an absolute measurement. • Ratio of prerequisite courses taken by s. Relative measure to show how much well-prepared the student is, in terms of the stated prerequisites. • Average terms past since prerequisite courses were taken by s.
(7) Performance relative to the course's level. (**c-perform**)	• The number of lower, same and higher level courses w.r.t. the level of c. • GPA over lower, same, higher level courses w.r.t. the level of c.
(8) Course-specific features. (**c-spec**)	• Course level that c belongs to. • Indicator whether c is in the student's major or not. • Average grade earned by past students.

The set C_{s,all_prior} represents the courses that the student took all the prior semester, before the target semester t. For any semester x, $C_{s,x}$ represents the set of courses that student s took on semester x.

Table 3: Statistics for the different classification tasks.

Task	Fgr	Wgr	RelF	RelCF	Agr
# instances	94,364	96,941	94,364	94,364	94,364
# positive	3,139	2,577	20,398	21,724	20,851
% positive	3.33	2.7	21.62	23.02	22.10

6.2 Performance analysis

Table 4 summarizes the performance of the various classification methods for the classification tasks, in terms of the AUC and the F_1 score. Based on both metrics, GB is the best performing method, closely followed by the RF classifier. As expected, DT, which is the simplest method, has the lowest performance. These results are better compared to the performance of grade prediction methods for any classification task. When using Course-Specific Regression for predicting the failing students, we get a F_1 score of 0.118, which is lower than any of the other methods we discuss.

While comparing the classification tasks, we can see that the tasks that predict relative performance have lower AUC values than when predicting absolute performance. In terms of F_1 scores, we can see clearly that the A-students are the most accurately predicted. The F_1 scores of the different tasks are related to the percentage of positive instances in each task. The tasks Fgr and Wgr, that are highly unbalanced, have significantly lower F_1 scores. Moreover, as there is 81% overlap between the students that are positive for both RelF and RelCF, the tasks of RelF and RelCF have very similar performance.

6.3 Feature importance study

One of our goals is to study which factors are important indicators of a student's performance, so we performed the

Table 4: Performance of the various classifiers.

Area under the ROC curve					
Classiffier	Fgr	Wgr	RelF	RelCF	Agr
DT	0.834	0.710	0.689	0.716	0.820
SVM	0.853	0.736	0.690	0.718	0.819
RF	0.873	0.778	0.748	0.759	0.850
GB	**0.877**	**0.780**	**0.755**	**0.765**	**0.854**
F_1 score					
Classifier	Fgr	Wgr	RelF	RelCF	Agr
DT	0.255	0.123	0.450	0.466	0.573
SVM	0.276	0.171	0.452	0.469	0.570
RF	0.317	0.165	0.499	0.502	0.604
GB	**0.319**	**0.181**	**0.506**	**0.507**	**0.610**

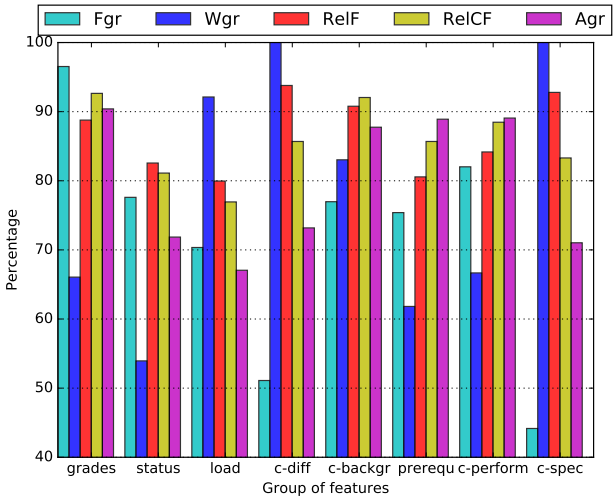

Figure 2: Percentage of performance managed to recover using only one group of features.

following experiment. We categorize each extracted feature to one of the the 8 groups, according to Table 1. Afterwards, for each classification task, we built RF classifiers using only the features belonging to one of the above groups. We selected to use RF over GB, as they achieve similar performance in less training time. The accuracy achieved for a model using a single group of features is expected to be less than the accuracy when using all the features. The percentage of accuracy that a model using only the features belonging to one group manages to achieve, in terms of the F_1 score, are presented on Fig. 2. In this bar chart, we can see the percentage of accuracy achieved from all the different feature groups for all the discussed classification tasks. The higher the percentage achieved by a single group of features, the more predictive ability these features have.

From this figure, we can get many insights on the factors that affect student performance. For example, the features related to the students' grades (group 1) have a very good predictive capability in almost all the tasks, except the task of predicting the W grades. In this task, features related with the course's difficulty and popularity (group 4) as well as features that are course-specific (group 8), manage to achieve the same accuracy as when using all the features. This indicates that the reasons that a student drops a course are related more to the course, rather than to the students themselves. The next best indicator is the feature group about the student's course load during the semester.

On the other hand, this is not the case for predicting the failing students, in the absolute sense, i.e., receive a D or F. When using only course-related groups (groups 4, 8) for predicting the students likely to fail a course (Fgr task), we manage to recover half or less from the F_1 score. As a result, these factors do not influence the absolute failing performance of a student, indicating that the reasons for that are mostly related with the student. As the students' grades manage to recover almost the same performance as when using all the features, they are the ones that affect the Fgr prediction the most. When using the other groups, it is very difficult to achieve comparable performance, as they recover 80% or less of the F_1 score.

The feature groups are behaving similarly for RelF and RelCF. However, we notice that for the RelCF task, the feature groups that are related with student-course specific features have slightly better performance, while the student-specific groups have slightly worst performance, compared to the task of RelF. This is happening because, for RelCF, we take into consideration how other students usually perform on the target course. Every single group has enough information for the RF to utilize to achieve performance which is as good as 75% of the best case, i.e., when using all the features.

Finally, for identifying the A-students, the feature groups 1, 5, 6, 7 are the ones that manage to have the best performance. These groups are related with students' grades in general, but also, with their grades relative to the target course's background, prerequisites and level. Using only one of them can provide us with the information we need in order to recover around 90% of the performance while using all the features.

7. CONCLUSIONS

The purpose of this paper is to accurate identify students that are at risk. These students might fail the class, drop it, or perform worst than they usually do. We extracted features from historical grading data, in order to test different simple and sophisticated classification methods based on big data approaches. The best performing methods are the Gradient Boosting and Random Forest classifiers, based on AUC and F_1 score metrics. We also got interesting findings that can explain the student performance.

8. ACKNOWLEDGEMENTS

This work was supported in part by NSF (IIS-1247632, IIP-1414153, IIS-1447788, IIS-1704074, CNS-1757916), Army Research Office (W911NF-14-1-0316), Intel Software and Services Group, and the Digital Technology Center at the University of Minnesota. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute.

9. REFERENCES

[1] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[2] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[3] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[4] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[5] Z. Iqbal, J. Qadir, A. N. Mian, and F. Kamiran. Machine learning based student grade prediction: A case study. *arXiv preprint arXiv:1708.08744*, 2017.

[6] J. E. Knowles. Of needles and haystacks: Building an accurate statewide dropout early warning system in wisconsin. *Journal of Educational Data Mining*, 7(3):18–67, 2015.

[7] S. Kotsiantis, C. Pierrakeas, and P. Pintelas. Predicting students'performance in distance learning using machine learning techniques. *Applied Artificial Intelligence*, 18(5):411–426, 2004.

[8] J. McFarland, B. Hussar, C. de Brey, T. Snyder, X. Wang, S. Wilkinson-Flicker, S. Gebrekristos, J. Zhang, A. Rathbun, A. Barmer, et al. Undergraduate retention and graduation rates. In *The Condition of Education 2017. NCES 2017-144*. ERIC, 2017.

[9] B. Minaei-Bidgoli, D. A. Kashy, G. Kortemeyer, and W. F. Punch. Predicting student performance: an application of data mining methods with an educational web-based system. In *Frontiers in education, 2003. FIE 2003 33rd annual*, volume 1, pages T2A–13. IEEE, 2003.

[10] S. Morsy and G. Karypis. Cumulative knowledge-based regression models for next-term grade prediction. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 552–560. SIAM, 2017.

[11] E. Osmanbegović and M. Suljić. Data mining approach for predicting student performance. *Economic Review*, 10(1):3–12, 2012.

[12] A. Pardo, N. Mirriahi, R. Martinez-Maldonado, J. Jovanovic, S. Dawson, and D. Gašević. Generating actionable predictive models of academic performance. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 474–478. ACM, 2016.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[14] A. Polyzou and G. Karypis. Grade prediction with models specific to students and courses. *International Journal of Data Science and Analytics*, 2(3-4):159–171, 2016.

[15] Z. Ren, X. Ning, and H. Rangwala. Grade prediction with temporal course-wise influence. In *Proceedings of the 10th International Conference on Educational Data Mining*, pages 48–55, 2017.

[16] C. Romero, S. Ventura, P. G. Espejo, and C. Hervás. Data mining algorithms to classify students. In *Educational Data Mining 2008*, 2008.

[17] A. A. Saa. Educational data mining & students' performance prediction. *International Journal of Advanced Computer Science & Applications*, 1:212–220, 2016.

[18] M. Sweeney, J. Lester, and H. Rangwala. Next-term student grade prediction. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 970–975. IEEE, 2015.

[19] M. Sweeney, H. Rangwala, J. Lester, and A. Johri. Next-term student performance prediction: A recommender systems approach. *arXiv preprint arXiv:1604.01840*, 2016.