Extracting Features to Classify Students Based on their Academic Performance

Mr. P. Subba Rao¹, B. Sweeya sahrudi², G. Srihari³, K. Karthikeya Chary⁴, S. Mahesh⁵
¹Assistant Professor, Malla Reddy Institute of Technology and Science, subbarao.p@mrits.ac.in
²Student, Malla Reddy Institute of Technology and Science, sweeyasahrudi35@gmail.com
³Student, Malla Reddy Institute of Technology and Science, gujarathisrihari02@gmail.com
⁴Student, Malla Reddy Institute of Technology and Science, karthikeyachary.k@gmail.com
⁵Student, Malla Reddy Institute of Technology and Science, mhshsanal2@gmail.com
Department of Computer Science & Engineering

Abstract
In today’s educational climate, developing tools to support students and learning in a traditional or online context is a crucial responsibility. The first stages in employing machine learning techniques to enable such technology centered on forecasting a student’s success in terms of marks earned. The disadvantage of these methods is that they are not as effective at predicting low-achieving students. The goal of our efforts is twofold. To begin, we investigate whether badly performing students may be more accurately predicted by recasting the task as a binary classification problem. Second, in order to learn more about the reasons that contribute to bad performance, we created a set of human-interpretable attributes that quantify these aspects. We conduct a study based on these characteristics to identify distinct student groups of interest while also determining their value.

Keywords: Binary classification, Performance, Human Interpretable Attributes

I. INTRODUCTION

Higher education institutions are always looking for ways to improve student retention and achievement. According to the National Center for Education Statistics in the United States 60% of undergraduate students pursuing four-year degrees will not complete their studies at the same institution where they began within the first six years. At the same time, 30% of first-year college students drop out after their first year. As a result, institutions are looking for more efficient and effective methods to serve students. This is where data mining comes in to help solve some of these challenges. Educational data mining and learning analytics were created to provide tools to aid the learning process, such as monitoring and measuring student progress, as well as predicting success and guiding intervention tactics.

The majority of present approaches are aimed at identifying students who are at risk of failing to complete a course or activity and who could benefit from additional support. One of the most important tasks in this procedure is to predict the student’s grade achievement. While reasonable prediction accuracy has been attained, the algorithms developed to identify low-performing pupils have a severe flaw. Typically, these models are overly optimistic about student performance because the vast majority of students perform well or satisfactorily.

We investigate the topic of predicting a student’s performance at the end of a semester before he or she actually enrolls in the course. The prediction problem is presented as a classification problem, in which two groups of students are created depending on their course performance, in order to focus on the students that require these systems the most. We can effectively distinguish between pupils who are likely to complete a course or activity successfully and those who appear to be struggling. Following the identification of the latter group, we may be able to provide additional resources and support to assist them in achieving their goals.

"Success" and "failure," on the other hand, can be relative or not. A B grade, for example, may be deemed a poor grade for an excellent student yet a good mark for a very poor student. We looked studied many methods for identifying groups of students in a course, including failing students, dropping the class, students performing worse than predicted, and students performing worse than expected while taking into account the course’s complexity.
We designed features that capture possible elements that influence grades at the conclusion of the semester in order to acquire a better understanding of the learning process and its most essential qualities. We present a complete analysis using these features to answer the following questions: Which characteristics are good predictors of a student’s success? Which characteristics are the most crucial? The results are intriguing since different traits are most essential for various classification tasks.

II. RELATED WORK

We will evaluate the associated research in this area because we are interested in estimating next-term student performance. The binary classification has been used to solve a variety of educational challenges, including predicting if a student would drop out of high school [6] and predicting whether a student will pass a module in a distance learning environment [7]. To offer a qualitative measure of students’ performance, multi-label classification was used. With data from a survey, decision tree and naive Bayes classifiers are employed in [17]. A learning management system’s attributes have been used to assess the outcome as Fail, Pass, Good, and Excellent [16], as well as to classify students [12]. Some approaches experiment with various ways to classify student performance.

Big data approaches have been used in the field of learning analytics in recent years, influenced by developments in recommender systems. Sweeney et al. [18] coined the term “next-term grade prediction” in the context of higher education, and it refers to the difficulty of forecasting each student’s grades in the courses that he or she will take during the next semester. The models SVD and factorization machines (FM) were put to the test. In a separate approach [15], students’ previous performance influences grade estimate in two ways while creating latent models. Additional state-of-the-art approaches, as well as a mix of FM and random forests, were applied in [19]. (RF). Historical grades and additional content attributes, which include student, course, and teacher characteristics, were used to generate the data. [14] and [10] developed course-specific algorithms based on linear regression and matrix factorization to predict next-term grades in the same situation.

Math

The ratio of true positives to all expected positives is known as precision. The ratio of true positives to all actual positives is known as recall. Precision refers to the classifier’s capacity to avoid labelling a negative sample as positive, whereas recall refers to the ability to locate all positive samples. The F1 score is a measurement of accuracy that is determined as follows:

\[ F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \]

Dataset

First, we’ll define several terminologies that are being used in this context. The performance of a student, s, in a course, c, at the end of the semester is referred to as an instance. Prior courses, marked by \( C_{\text{all prior}} \), are all the courses that a student took in previous semesters before taking course c. \( C_s \) denotes the set of courses for a single semester x. Furthermore, for a course c, there may be a set of prerequisites that a student must complete before attempting c. This group of courses is referred to as the prerequisites. Every course \( x \), cr, is worth a certain amount of credits.

In order to graduate from a college or university, an undergraduate student must take at least one course per semester and achieve a good mark. These courses may be mandatory, electives, or just courses that the student takes for his or her own growth, intellectual curiosity, or fun, depending on the student’s degree program. If a student drops out of a course after the first two weeks of classes, the letter ‘W’ appears on their transcript.

The original dataset, which spans 13 years, was received from the University of Minnesota. Any instances of a letter grade that did not fall inside the A–W grading scale were eliminated to grades (A+, A, A–, B+, B, B–, C+, C, C–, D+, D, D–, F, W). The letter grade A is the most common in our database. For the last ten autumn and spring semesters, we extract features for the events that occurred. We use all of the students who have previously taken the course in a semester, and we extract a set of attributes for each student attending the course. We also create features for instances assigned the letter W, but we don’t use them in any other manner during the feature extraction process. These will only be utilized when attempting to anticipate the number of students who will drop out of a course.
ARCHITECTURE

Figure 1: Classification Architecture

Data preprocessing is a data mining strategy that entails converting raw data into a comprehensible format. Separating data into training and testing sets is an important part of evaluating data mining algorithms, as shown in Fig 1. After separating the data into training and testing sets, the set with the majority of the data is used for training, while the set with the minority of the data is used for testing.

Extracted Features
We extracted different attributes from prior grading data to capture plausible explanations for a student’s low performance. Student specific features (independent of course c), course-specific features (independent of student s), and student- and course specific features (they are a function of both s and c) are the three types of features. Tables 1 and 2 contain descriptions of all extracted features, with relevant traits grouped together into eight different groups. Later on, the bold keywords will be used to indicate the corresponding group of features. Note that we construct a separate set of features for each s, t, c, where student s took course c in semester t. A student’s attempt to take course c at a certain moment in his or her studies is characterized by a variety of characteristics.

Numerical, category, or indicator variables are examples of these characteristics. We utilize the values 0 or 1 for indicator features. A numerical value is used to encode categorical information. The current semester feature, for example, is categorical, and the values fall, spring, and summer are changed to 0,1,2, respectively.

Existing System
In today's educational climate, creating apparatuses that assist understudies and studying in a traditional or online context is a major undertaking. The first steps toward enabling such advancements using AI processes focused on predicting the understudy’s presentation in terms of completed evaluations.

Disadvantages
- There are no effective techniques to classify students based on their academic performance.
- Due to restricted internet setting or connectivity, it was a difficult task to obtain effective results.
- Do not perform well in predicting poor performing students.

Proposed System
Our work has a two-fold objective. To begin, we look into whether understudies that are ineffectively per-framing might be more clearly predicted by describing the problem as twofold characterization. Second, we created several human interpretable highlights that evaluate these elements in order to gain bits of knowledge about which elements can cause poor showing. The understudies’ assessments from the University of Minnesota, a public undergrad establishment, were used to compile these characteristics.
Advantages
- Used machine learning algorithms which include Support vector machine, Random forest algorithm, gradient boosting, Decision tree algorithm for classifying students based on their academic performance effectively.
- Large amount of data can be processed at the same time.

Classification Algorithms

A. Support Vector Machine
SVM (Support Vector Machine) is a supervised machine learning technique for solving classification and regression issues. It is, however, mostly used to tackle categorization problems. In this technique, each data item as a point in n-dimensional space, the value of each feature is the value of a certain coordinate (where n is the number of characteristics you have). Then we locate the hyper plane that best distinguishes the two classes to complete classification (look at the below snapshot). Support Vectors are individual observation coordinates. The Support Vector Machine (SVM) is a frontier that does the greatest job of distinguishing between the two classes (hyper-plane and line). A support vector machine, in more technical terms, creates a hyper plane or series of hyper planes in a high- or infinite dimensional space that can be used for classification, regression, or other tasks such as outlier detection. Intuitively, the hyper plane with the greatest distance to the nearest training-data point of any class (so-called functional margin) achieves a decent separation, because the larger the margin, the smaller the classifier’s generalization error. Even if the initial problem is expressed in a finite dimensional space, the sets to discriminate are frequently not linearly separable in that space. As a result, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, so facilitating separation.

B. Random Forest Algorithm
It’s an ensemble algorithm, which means it’ll develop an accurate classifier model by combining numerous classifier algorithms inside. Internally, this technique will construct a train model for classification using a decision tree algorithm. Random Forest, a well-known machine learning algorithm. It can be used to predict the results for classification and regression problems in machine learning. It is the process of combining multiple classifiers to solve a complicated problem and increase the performance of the model. The random forest collects data from each tree and predicts the final output based on the majority votes of projections.

A. Decision Tree Algorithm
By arranging all similar data in the same branch of the tree, the decision tree algorithm will develop a training model, and this process will continue until all records are arranged in the entire tree. The classification train model refers to the entire tree.

B. Gradient Boosting Algorithm
GRADIENT BOOSTING ALGORITHM Gradient boosting classifiers are a collection of machine learning algorithms that combine several weak learning models to generate a powerful predictive model. When doing gradient boosting, decision trees are commonly employed. Gradient boosting models are gaining popularity as a result of their ability to categorise difficult datasets, and they have lately won a number of Kaggle data science challenges. ScikitLearn, a Python machine learning package, provides a variety of gradient boosting classifier implementations, including XGBoost. A single accurate train model will be built by combining different techniques. Gradient Boosting improves performance in all of these techniques.

Experiments

A. Experimental design
The models created are global, meaning that a single model may predict several outcomes. The overall achievement of all students across all courses All Any time a student takes a test, features are extracted. A program Because randomization is used in the models, in order to sample and/or initialize the model, we run it with 5 different seeds and average out the results. For classifier evaluation, we used cross validation. The information are divided into five distinct subsets. Test each fold on a piece of paper. One partition will be used for training, while the others will be used for storage. The average of the five folds’ evaluation metrics will be the reported values.

Metrics. The ratio of true positives to all predicted positives is known as precision. The percentage of true positives to all true positives is known as recall. A classifier should not categorize a negative sample as positive. Recall refers to the ability to locate all positive samples. F1 score is a measurement of accuracy that is calculated as follows:
\[ F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \]

The area under the receiver operating characteristic (ROC) curve, also known as AUC, is used to measure a classifier's performance across all thresholds. At varying thresholds, the ROC curve shows the true positive rate against the false positive rate. The AUC indicates the likelihood that a random positive instance will be ranked higher than a negative one by the classifier.

1. Calculating the positive threshold, we can assign a prediction score in the range of [0,1] to a test instance instead of a label. AUC is a metric that can be computed. To determine a criterion for the above a certain prediction score, we take the following steps in positive class: Sort the predictions first. The scores are listed in a non-increasing order.
2. Calculate the \( F_1 \) score for each point \( L \) in this sorted sequence using Eq. 1, assuming that all instances with a prediction score greater than the \( L \)th instance are classed as positive and all others are categorized as negative.
3. The maximum \( F_1 \) value attained above is the \( F_1 \) score.

**B. Performance analysis**

In terms of the AUC and the \( F_1 \) score Table 1 highlights the performance of the various classification methods for the classification tasks. On both metrics, the GB technique is the top performer, followed by the RF classifier.

The simplest method, DT, has the lowest performance, as expected. When compared to the performance of grade prediction algorithms for any classification problem, their results are superior. We get an \( F_1 \) score of 0.118 when using Course-Specific Regression to predict failing students, which is lower than any of the other methods we discuss.

When comparing the classification tasks, it is clear that those that predict relative performance have lower AUC values than those that predict absolute performance.

We can easily observe that the A-students are the most accurately predicted in terms of \( F_1 \) scores. The percentage of good instances in each task is related to the \( F_1 \) scores of the various tasks. \( F_1 \) scores are considerably lower for the severely imbalanced tasks \( F_{gr} \) and \( W_{gr} \). Furthermore, because there is an 81 percent overlap in students who are positive for both RelF and RelCF, the tasks of RelF and RelCF function similarly.

**C. Feature Importance Study**

We conducted the following experiment to learn which factors are crucial markers of a student's performance, which is one of our goals. Afterwards,

<table>
<thead>
<tr>
<th>Table 1: Performance of various classifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area under the ROC curve</td>
</tr>
<tr>
<td>Classifier</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>DT</td>
</tr>
<tr>
<td>SVM</td>
</tr>
<tr>
<td>RF</td>
</tr>
<tr>
<td>GB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>F_{1} score</th>
<th>F_{gr}</th>
<th>W_{gr}</th>
<th>RelF</th>
<th>RelCF</th>
<th>A_{gr}</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>0.255</td>
<td>0.123</td>
<td>0.450</td>
<td>0.466</td>
<td>0.573</td>
</tr>
<tr>
<td>SVM</td>
<td>0.276</td>
<td>0.171</td>
<td>0.452</td>
<td>0.469</td>
<td>0.570</td>
</tr>
<tr>
<td>RF</td>
<td>0.317</td>
<td>0.165</td>
<td>0.499</td>
<td>0.502</td>
<td>0.604</td>
</tr>
<tr>
<td>GB</td>
<td>0.319</td>
<td>0.181</td>
<td>0.506</td>
<td>0.507</td>
<td>0.610</td>
</tr>
</tbody>
</table>

**Figure 1: Percentage of performance managed to recover only one group of features.**
We created RF classifiers for each classification job using only one of the above-mentioned groups of characteristics. We chose RF over GB since they produce similar results with less training time. It is assumed that level of precision attained by a model based on a single set of features will be less complex. On Fig. 1, the percentage of accuracy that a model using only characteristics from one group is able to achieve in terms of the F1 score is shown. We can see the percentage of accuracy gained from all of the different feature groups for all of the classification tasks described in this bar chart. The higher the percentage reached by a particular group of features, the better these features’ predictive power. We can learn a lot about the elements that influence student success by looking at this graph. For example, variables associated to students’ grades exhibit excellent predictive power in practically all tasks, with the exception of predicting W grades.

In this test, features linked to the course’s difficulty and popularity and course-specific features obtain the same accuracy as when all features are used. This suggests that the reasons students leave a course have more to do with the course than with the students themselves. The feature group about the student’s course load during the semester is the next best indicator. On the other hand, this is not the case when it comes to forecasting failing pupils in the strictest sense, i.e. those that score a D or F. We recover half or less of the F1 score when just course-related groupings are used to predict students who are likely to fail a course.

As a consequence, these characteristics have little bearing on a student’s absolute failing performance, implying that the reasons behind this are mostly tied to the student. Because the students’ grades regain almost the same performance when all of the characteristics are included, they have the greatest impact on the Fgr prediction. When employing the other groups, it’s difficult to attain comparable results because they only recover about 80% of the F1 score.

For RelF and RelCF, the feature groups behave similarly. However, when comparing the RelCF work to the RelF job, we see that the feature groups associated to student-course specific characteristics perform somewhat better, while the student-specific groups perform slightly worse.

This is because, while using RelCF, we examine how other students typically perform on the target course. Every single group contains enough data for the RF to use to produce performance that is at least 75% of the best scenario, i.e. when all features are used. Even if we only use one of them, we can get the information we need to regain roughly 90% of the performance when employing all of the features.

III. Conclusion

The goal of this article is to accurately identify at-risk students. These pupils may fail the class, withdraw from it, or perform poorly than usual. We used historical grading data to extract characteristics in order to test several simple and advanced classification methods based on big data approaches. Based on AUC and F1 score measures, the highest performing algorithms are Gradient Boosting and Random Forest classifiers. We also discovered some intriguing findings that help to explain student performance.

References

ith temporal course
buted denial of service attacks using an adaptive spin lock rate control
arning Research
n. In [29]
[28]
[27]
[26]
[25]
[24]
[23]
[22]
[21]
[20]
[19]
[18]
[17]
[16]
[15]
[14]
[13]
[12]
[11]
[10]
[9]

Mr. P. Subba Rao, B. Sweeya sahrudi, G. Srihari, K. Karthikeya Chary, S. Mahesh


[25] Dr. V. Senthil kumar, Mr. P. Jeevanantham, Dr. A. Viswanathan, Dr. Vignesh Janarthanan, Dr. M. Umamaheswari, Dr. S. Sivaprakash Emperor Journal of Applied Scientific Research “Improve Design and Analysis of Friend-to-Friend Content Dissemination System “Volume - 3 Issue - 3 2021


