Learning Recommendations from Educational Event Data in Higher Education

Gyunam Park, Lukas Liss and Wil M. P. van der Aalst

Chair of Process and Data Science, RWTH Aachen University, Ahornstraße 55, Aachen, 52074, Germany.

*Corresponding author(s). E-mail(s): gnpark@pads.rwth-aachen.de;
Contributing authors: liss@pads.rwth-aachen.de;
wvdaalst@pads.rwth-aachen.de;

Abstract

This paper presents a novel approach for generating actionable recommendations from educational event data collected by Campus Management Systems (CMS) to enhance study planning in higher education. The approach unfolds in three phases: feature identification tailored to the educational context, predictive modeling employing the RuleFit algorithm, and extracting actionable recommendations. We utilize diverse features, encompassing academic histories and course sequences, to capture the multi-dimensional nature of student academic behaviors. The effectiveness of our approach is empirically validated using data from the computer science bachelor's program at RWTH Aachen University, with the goal of predicting overall GPA and formulating recommendations to enhance academic performance. Our contributions lie in the novel adaptation of behavioral features for the educational domain and the strategic use of the RuleFit algorithm for both predictive modeling and the generation of practical recommendations, offering a data-driven foundation for informed study planning and academic decision-making.

Keywords: Educational Recommendations, Study Planning, Educational Event Data, Higher Education, Rule-based Modeling, Recommender Systems

1 Introduction

In higher education, the design of study programs is a complex task that involves the careful consideration of examination regulations, degree requirements, grading criteria, thesis guidelines, and course cancellation policies [1]. These elements serve as the foundation for guiding students through their academic journey and ensuring that the curriculum meets educational standards and student needs. For example, the examination regulation for the 2018 RWTH bachelor's program in computer science mandates that students must accumulate at least 120 credit points to register for their thesis, while the course handbook outlines prerequisites for course enrollment, like completing "Introduction to Computer Engineering" course before advancing to "Practical System Programming" course. These structured requirements assist students in tailoring their course selections to their interests and academic objectives and ensuring that courses are not taken too early.

The development of effective study programs includes creating recommended study plans [2]. These plans suggest a sequence for completing courses that help students graduate on time. This underscores the importance of providing study program designers with the knowledge and tools needed to create curricula that are both academically sound and flexible enough to accommodate various student needs.

One approach to assist program designers for creating study plans is to analyze historical data on student study paths [3–7]. This analysis can identify patterns associated with successful academic outcomes, offering a data-driven basis for curriculum development. By using these insights on what has helped students succeed in the past, program designers can create study plans that include recommended course selection and sequencing.

However, existing work mainly analyzes how individual courses or grades affect academic performance, but often misses the sequence and interaction between courses. To fill this gap, the authors in [8] investigate students' course trajectories, considering the sequence between courses, for better course sequencing guidance. The proposed method uses various features based on course sequencing such as directly-follows relations between courses and employ decision trees to derive course sequences that contribute to better academic performance. However, the method only considers control-flow-centric features about course sequencing, limiting its ability to offer holistic recommendations from different perspectives. Moreover, the proposed approach inherits issues of decision trees such as overfitting and outlier sensitivity, providing less practical recommendations.

In this work, we extend the approach suggested in [8] by systematically identifying behavioral features that mirror educational processes. To address decision tree drawbacks, we apply the RuleFit algorithm, ensuring the interpretability vital for creating effective recommendations. More in detail, we introduce an approach for deriving such recommendations from educational event data sourced from Campus Management Systems (CMS). Our approach involves three phases for learning recommendations from educational event data:

- First, we identify behavioral features that encapsulate students' academic histories, such as previous grades, course sequences, and exam attempt frequencies. To

achieve this, we explore a wide array of features suggested, specifically leveraging the extensive list proposed by de Leoni et al. [9]. We then tailor these features to the educational domain, ensuring they are relevant and reflective of students' academic journey.
- After extracting these features from the event data captured by CMS, we train a predictive model using the RuleFit algorithm [10]. This model is designed to forecast educational outcomes such as overall GPA, time-to-degree, and the likelihood of dropout, providing a statistical foundation for recommendation generation.
- Finally, with the predictive model, we extract actionable rules distilled from the model's outputs and transform these rules into recommendations for course designers.

We evaluate the effectiveness of our proposed method through an empirical evaluation using educational event data from the computer science bachelor's program at RWTH Aachen University. We first extract relevant features for 3,190 students within the program. Next, we develop a predictive model specifically designed to forecast students' GPA using the features. Finally, we extract rules statistically associated with achieving the goal from the predictive model and derive recommendations for the computer science bachelor's program by analyzing these rules.

The remainder of this paper is structured as follows. Section 2 reviews related literature, and Section 3 introduces the preliminaries necessary for understanding the context and foundation of our work. Our main contributions are elaborated in Section 4, detailing our proposed approach. Next, we describe the dataset employed in our evaluation in Section 5. Afterward, Section 6 presents a comprehensive evaluation of our approach, including the results and their interpretation. Finally, Section 7 concludes the paper.

2 Related Work

Student success is essential for higher education institutions, serving as a key metric for evaluating their quality. York et al. [11] define student success, emphasizing seven core elements: academic achievement, satisfaction, skills, and competencies acquisition, persistence, learning objectives attainment, and career success. Many studies predominantly equate student success with academic achievement, measured by Grade Point Average (GPA), time-to-degree, and dropout [12].

Identifying the factors that influence student success necessitates gathering and analyzing relevant data. This includes historical academic performance, such as high school performance and pre-admission scores, alongside university data like GPA and specific course grades [13]. Student demographics also play a role, encompassing gender, age, ethnicity, and socioeconomic background [14]. Moreover, psychological factors like interest, study habits, stress levels, and motivation can impact outcomes [15]. Lastly, student E-Learning activity metrics, such as login frequency, task completion, test participation, forum contributions, and engagement with educational content, are critical for a comprehensive understanding [16].

A variety of methods have been introduced to provide recommendations to students or course designers by correlating the aforementioned factors with academic achievement. These approaches leverage data and process mining techniques [17–19]. A significant emphasis is placed on E-Learning activities. For instance, sequential pattern mining has been employed to customize recommendations on learning materials according to students' learning styles and web-usage habits [20]. Association rule mining has been utilized to suggest online learning activities on educational websites [21], and for personalizing content recommendations by analyzing web browsing events with educational context [22]. Furthermore, clustering techniques have been applied to develop a model for recommending resources to students in analogous situations [23], and to personalize E-Learning by organizing web documents using clustering methods based on maximal frequent item sets [24].

Focusing on methods that correlate historical academic performance with academic success, various techniques have been employed to predict and enhance student outcomes. Decision trees and genetic algorithms have been used to identify student dropout risks, revealing that lower GPAs and extended enrollment are key indicators [3]. Similarly, random forest methods have identified significant factors affecting dropout rates, utilizing a dataset of informatics engineering students to assess the impact of historical academic features [4]. Predictive models employing support vector machines and k-nearest neighbor algorithms have been developed in [5] to forecast final exam grades, with a focus on midterm grades and academic departments as predictors. Early assessment activities have been used to identify students at risk in introductory courses, applying random forest techniques for timely interventions [6]. Furthermore, Alangari and Alturki [7] explored GPA predictions, identifying specific courses as significant academic performance influencers.

The existing work on providing recommendations based on historical academic performance primarily examine the influence of individual courses or grades on academic achievement, with limitations in addressing the sequence and interaction of courses. To overcome this, the authors in [8] investigate students' study paths across multiple courses, considering the effects of retaking courses to aid students in course selection. However, this approach, with its reliance on ad-hoc, control-flow-focused features, falls short of generating comprehensive recommendations. Furthermore, using decision trees, prone to overfitting and sensitivity to outliers, raises concerns about robustness. This work enhances this approach by systematically developing actionable behavioral features that represent educational processes. We also mitigate decision tree limitations by employing the RuleFit algorithm, ensuring interpretability essential for generating effective recommendations.

3 Preliminaries

This section outlines the foundational concepts of our approach. We start with an overview of educational event data, which form the basis of our analysis. Following this, we introduce the *RuleFit* algorithm, a key method for developing predictive models from which we derive recommendations.

3.1 Educational Event Log

Table 1 illustrates a fragment of an educational event log, presented in a tabular format, which serves as an input for our approach. Each row within the table represents a discrete event. An event denotes the activity of a student taking a course in their study program. Events in the log are characterized by a set of attributes. In this paper, we focus on six attributes for each event:

- *Student ID*: A unique identifier for each student.
- *Course ID*: The identifier of the course.
- *Semester*: The academic term during which the course was taken, indicating the temporal aspect of the student's academic journey.
- *Exam Date*: The date when the course exam was taken.
- *Credits*: The credit value of the course.
- *Grade*: The grade achieved by the student in the course.

Student ID	Course ID	Semester	Exam Date	Credits	Grade
S001	C101	Fall 2023	15.Dec.2023	4	2.0
S001	C102	Spring 2024	18.Jun.2024	3	1.5
S001	C101	Fall 2024	15.Dec.2024	4	1.7
S001	C104	Fall 2024	19.Dec.2024	3	1.3
S002	C101	Fall 2023	15.Dec.2023	4	1.0
S002	C103	Spring 2024	15.Jun.2024	3	2.3
S003	C102	Fall 2023	15.Dec.2023	3	2.7
S003	C101	Spring 2024	18.Jun.2024	4	2.0

Table 1: Example of educational event data

For instance, the first row indicates that student S001 took the course C101 in the Fall 2023 semester, for which the student earned 4 credits and received a grade of 2.0. For the remainder of the paper, we assume that 1.0 is the highest (best) grade and 5.0 is the lowest (worst) grade. Students with a grade of 4.0 or better pass the exam.

3.2 The RuleFit Algorithm

The RuleFit algorithm is a machine learning technique that synergizes the rule-based decision-making power of decision trees with the predictive accuracy and interpretability of linear models. Unlike traditional decision tree models, which often struggle with overfitting and limited interpretability due to complex tree structures, RuleFit simplifies the decision process and enhances model understandability without sacrificing performance [10]. The algorithm operates in three phases: rule generation, rule fitting, and feature importance.

3.2.1 Rule Generation

In the first phase, the algorithm creates decision rules from an ensemble of trees, which are constructed using gradient boosting [25]. Each rule is a conditional statement that mirrors a path from the root node to a leaf node within these trees. For example, take features such as the course grade ($l_{\text{grade\_course-name}}$) and the number of course enrollments ($l_{\text{num\_course-name}}$). A rule that might be generated (r_1) could be: IF $l_{\text{grade\_C101}} < 1.3$ AND $l_{\text{num\_C101}} < 2$ THEN predict the overall GPA of 1.5. This rule suggests that achieving a high grade for C101 with a single attempt is associated with higher academic achievement, which is represented by the predicted overall GPA.

3.2.2 Sparse Linear Model with a Running Educational Example

After generating a set of rules, RuleFit constructs a sparse linear model that includes both these rules and the original features from the dataset. For our educational example, this means integrating features like $l_{\text{grade\_C101}}$ and $l_{\text{num\_C101}}$ with rules based on them like r_1.

In the RuleFit algorithm, the sparse linear model is represented as follows:

$$\hat{f}(x) = \hat{\beta}_0 + \sum_{k=1}^{K} \hat{\alpha}_k r_k(x) + \sum_{j=1}^{p} \hat{\beta}_j l_j(x_j),$$

where:

- $\hat{f}(x)$: The predicted outcome based on the model, such as a student's overall GPA.
- $\hat{\beta}_0$: The intercept of the model.
- K: The total number of rules generated from decision trees.
- $\hat{\alpha}_k$: The weight assigned to the k-th rule.
- $r_k(x)$: The k-th rule applied to the input data x.
- p: The number of original features in the dataset.
- $\hat{\beta}_j$: The weight assigned to the j-th original feature.
- $l_j(x_j)$: The j-th original feature in the input data x.

Lasso regularization [26] is applied to prune less informative rules and features, focusing the model on those most predictive of the outcome.

3.2.3 Determining Feature Importance

The final phase quantifies the importance of each feature and rule. For determining feature importance, the importance measure for original features is defined as:

$$I_j = |\hat{\beta}_j| \cdot \text{std}(l_j(x_j)),$$

and for rules as:

$$I_k = |\hat{\alpha}_k| \cdot \sqrt{s_k(1 - s_k)},$$

where:

- $\text{std}(l_j(x_j))$: The standard deviation of the j-th original feature across the dataset, reflecting its variability.
- s_k: The support of the k-th rule, representing the proportion of data instances where the rule is applicable.

These measures help identify which features and rules are most strongly associated with the outcome.

4 Approach to Learning Recommendations

Figure 1 provides an overview of our approach to learning actionable study planning recommendations from educational event data extracted from CMS. First, we define objectives for our recommendations, such as improving overall GPA and optimizing course completion metrics. Next, we embark on feature engineering, where we apply established frameworks to define a comprehensive set of behavioral features that capture students' academic behaviors and experiences. With our features in place, we transition to the model construction phase, where the RuleFit algorithm is employed to train an ensemble of decision trees on the engineered features. Using the constructed model, we generated recommendations by assessing the importance and impact of the rules that constitute the model and prioritizing those with a higher influence on academic success. Subsequently, we interpret these recommendations to ensure they are actionable, aligning them with real-world educational contexts and objectives.

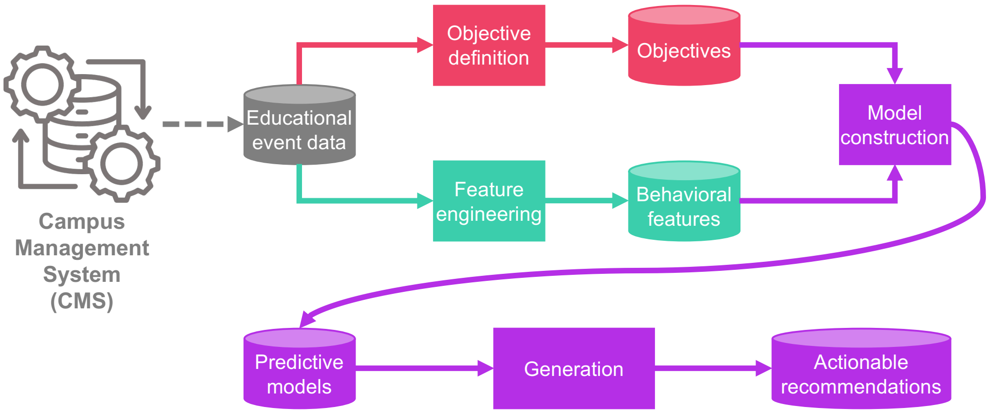


Figure 1: Overview of the approach.

4.1 Objective Definition

The objectives of our recommendations are defined by establishing metrics that encompass various dimensions of academic performance. These metrics are divided into two categories: those pertaining to overall study progress and achievements, and those focused on performance in individual courses.

4.1.1 Study-Level Performance Metrics

For evaluating performance at the study level, we consider the following metrics:

- **Overall GPA**: The cumulative Grade Point Average (GPA) serves as a comprehensive indicator of a student's academic performance across all courses undertaken during their complete study. For example, in the grading system considered in our work, a *lower* overall GPA is reflective of consistent academic excellence, with 1.0 being the highest (best) grade and 5.0 indicating failure.

- **GPA up to N-th Semester (GPA-N)**: This metric calculates the cumulative GPA of a student up to the N-th semester, providing a snapshot of academic performance in varying stages of the degree program.
- **Time-to-Degree**: This metric measures the duration taken by a student to complete their degree program, from enrollment to graduation. Shorter times-to-degree are generally indicative of efficient academic progression, whereas longer durations may signal potential challenges or changes in academic direction.
- **Dropout**: This metric assesses whether a student discontinues their studies before completing the degree requirements.

4.1.2 Course-Level Performance Metrics

For the performance evaluation at the individual course level, we consider the following metrics:

- **Course Grade**: The grade achieved in a particular course, which directly reflects a student's understanding of the subject matter.
- **Course Repeat**: The number of course attempts by the student. This metric can highlight issues such as course withdrawals or failures.

By defining these metrics, we establish a clear set of objectives for our recommendations to enhance both broad and specific aspects of student academic performance. These objectives guide the development and evaluation of our predictive models and the recommendations derived from them, ensuring that they are aligned with meaningful academic outcomes.

4.2 Feature Engineering

Following the framework suggested by de Leoni et al. [9], we categorize behavioral features into various perspectives, such as control-flow, data, and time perspectives, to characterize the educational event data comprehensively. Although the resource perspective is less applicable in our context, the other dimensions provide a foundation for analyzing student academic pathways and performance.

4.2.1 Control-Flow Perspectives

Control-flow features reflect the academic progression of students by capturing the sequence and occurrence of course enrollments:

- *Occurrences* (*occur*): The number of times a student is enrolled in a particular course. For instance, student S001 has enrolled in course C101 two times, i.e., $occur(C101) = 2$.
- *First Occurrence of a Course* (*first*): The first semester a student is enrolled in a course. Student S001's first enrollment in course C101 was in the first semester for the student (Fall 2023), i.e., $first(C101) = 1$.

- *Last Occurrence of a Course* (*last*): The most recent semester a student is enrolled in a course. Student S001's last enrollment in course C101 was in the third semester for the student (Fall 2024), i.e., $last(C101) = 3$.
- *Directly-Follows* (*DF*): A sequence where a student enrolls in one course in a semester and another course in the immediately following semester. For example, student S001 took course C101 in Fall 2023 and then took course C102 in Spring 2024, i.e., $DF(C101, C102) = True$.
- *Eventually-Follows* (*EF*): A sequence where a student takes one course and then another course in any subsequent semester. For example, student S001 took course C101 in Fall 2023 and then took course C104 in Fall 2024, with another semester in between, i.e., $EF(C101, C104) = True$.

4.2.2 Data Perspectives

Data features concentrate on event-related attributes such as grades and credits:

- *Total Credits* (*total_credits*): Sum of all credits accumulated by a student throughout their academic journey. For S001, this would be 4 (from C101 in Fall 2023) + 3 (from C102 in Spring 2024) + 4 (from C101 in Fall 2024) + 3 (from C104 in Fall 2024) = 14 credits, i.e., $total\_credits = 14$.
- *Total Credits in a Semester* (*total_credits_semester*): The total number of credits a student accumulates in a single semester. For S001's third semester in Fall 2024, the cumulative credits are 4 (from C101) + 3 (from C104) = 7 credits, i.e., ($total\_credits\_3 = 7$).
- *Average Grade* (*avg_grade*): The average grade across all courses for a student. For S001, the average grade is (2.0 + 1.5 + 1.7 + 1.3) / 4 = 1.625, i.e., $avg\_grade = 1.625$.
- *Latest Recorded Course Grade* (*course_grade*): The most recent grade received for each course. For S001, the latest recorded grade for C101 is 1.7 in Fall 2024, i.e., $course\_grade(C101) = 1.7$.

4.2.3 Time Perspectives

Time features capture the temporal aspects of academic activities:

- *Study Duration* (*study_duration*): The total number of semesters a student has been enrolled. S001 has been enrolled for 3 semesters, i.e., $study\_duration = 3$.
- *Course Duration* (*course_duration*): The number of semesters it takes for a student to complete a course. For S001, if we assume that C101 required two attempts, then the course duration for C101 is 2 semesters, i.e., $course\_duration(C101) = 2$.
- *Duration between Two Courses* (*duration_between*): The number of semesters between a student's enrollments in two courses. For S001, the duration between C101 (Fall 2023) and C102 (Spring 2024) is 1 semester, i.e., $duration\_between(C101, C102) = 1$.

By integrating these diverse perspectives, we develop a comprehensive feature set that encapsulates the nature of student academic behaviors. Note that the overlap in metrics across different perspectives is by design. Each perspective serves a unique analytical purpose despite superficial overlaps. For example, *Last Occurrence of a Course* in the control-flow perspective helps in pinpointing the specific timing within the student's academic trajectory, which differs from *Course Repeat*, which indicates the student's progression in the course. Similarly, while *Directly-Follows* and *Eventually-Follows* might seem to duplicate the sequence information, they provide distinct insights; the former captures immediate sequential dependencies, whereas the latter encompasses long-term course progressions. This differentiation aids in constructing a more detailed and comprehensive predictive model. Furthermore, by employing techniques such as *Lasso* regularization within the RuleFit algorithm, we manage the potential for redundancy and ensure that each feature contributes uniquely to the predictive accuracy.

4.3 Recommendation Distillation

Next, we distill recommendations from educational event data in two steps. First, we construct a predictive model to correlate behavioral features extracted from the educational event data to an objective. Second, we generate recommendations by analyzing the model's output, emphasizing the importance and coefficients of features and rules to identify key influencers of academic success. Subsequently, we interpret these statistical insights into practical strategies, aiming to enhance student outcomes by contextualizing the data-driven findings into actionable educational recommendations.

4.3.1 Model Construction

Using the objectives defined in Subsection 4.1 and the behavioral features outlined in Subsection 4.2, we employ the RuleFit algorithm to train an ensemble of trees. The RuleFit algorithm is particularly suited for this task due to its ability to generate interpretable models that combine the predictive power of decision trees with the simplicity of linear models. This hybrid approach allows for extracting actionable rules from complex educational data, capturing both linear relationships and decision-based patterns.

The selection of features depends on the chosen objectives. For an objective like overall GPA, we can incorporate features that are predictive yet independent of the GPA itself to avert data leakage. These features include *First Occurrence of Course X*, *Last Occurrence of Course X*, *Directly-Follows*, and *Cumulative Credits in a Semester*. Conversely, we consciously omit features closely tied to the GPA, such as *Average Grade* or *Latest Recorded Grade*.

4.3.2 Recommendation Generation

The resulting RuleFit model provides two key metrics for each feature and rule: the importance and the coefficient. We start by assessing the importance measure, which reflects the relative contribution of each feature or rule to the model's predictive accuracy. Features and rules with higher importance scores are considered more critical for prediction. This allows us to pinpoint the most influential elements that significantly affect student outcomes. As depicted in Table 2, we give precedence to rules with higher importance scores, as they are more impactful in the model.

Rule	Importance	Coefficient
$total\_credits\_1 \leq 15$	0.90	-0.15
$first(C101) = 1$	0.80	-0.18
$last(C103) = 4$ & $DF(C102, C103)$	0.70	0.20

Table 2: Top rules derived from the RuleFit model for predicting overall GPA

Subsequently, we analyze the coefficients to understand the nature and extent of each rule's impact. In a grading system where a lower GPA is indicative of better performance, a negative coefficient is desirable as it suggests that adherence to the rule is associated with an improvement in overall GPA. The magnitude of these coefficients provides insight into the strength of the influence that each feature or rule has on academic performance. For instance, the first rule in Table 2 regarding the limit on total credits in the first semester has a negative coefficient of -0.15, implying that students who take a moderate number of credits are likely to achieve better GPAs. In contrast, the third rule, as seen in Table 2, carries a positive coefficient of 0.20. This suggests that the last occurrence of C103 in the fourth semester, coupled with it directly following C102, is related to a higher (worse) GPA.

Finally, through contextual interpretation, we derive actionable recommendations to improve student performance. For instance, based on the first rule in Table 2, we recommend study program designers balance students' course loads, e.g., to the maximum of 15 credits, particularly in the initial semester, to enhance their academic outcomes. Additionally, according to the third rule, we recommend program designers carefully consider the order in which C103 and C102 are taken, especially if C103 is planned for the fourth semester.

5 Dataset Description and Scoping

This section introduces the event data used for evaluating our proposed approach. The event data are extracted from the CMS of RWTH Aachen University and are filtered for exam attempts by computer science bachelor students at RWTH Aachen University. Only students enrolled in the 2018 exam regulation for computer science bachelor's are included in the dataset. When the 2018 exam regulation was introduced, students from earlier semesters were allowed to change from older exam regulations to the one of 2018. Therefore, the event data comprises students who started earlier than 2019 but switched to the 2018 exam regulations and students who started from 2019 onwards. This ensures consistency in the features used for our predictive model, as all students' data reflect the revised course credit points and available course selections under the 2018 regulations.

In total, 78,529 exam attempts of 3,190 students from September 2000 to Januar 2023 are contained in the dataset. The event data follow the format described in Subsection 3.1. The student IDs are anonymized. Nearly every study path is unique, with an average number of 24.6 exam attempts per student which counts for an average of 110.8 credits points that students enrolled for. In total, there are 3,051 unique study path variants.

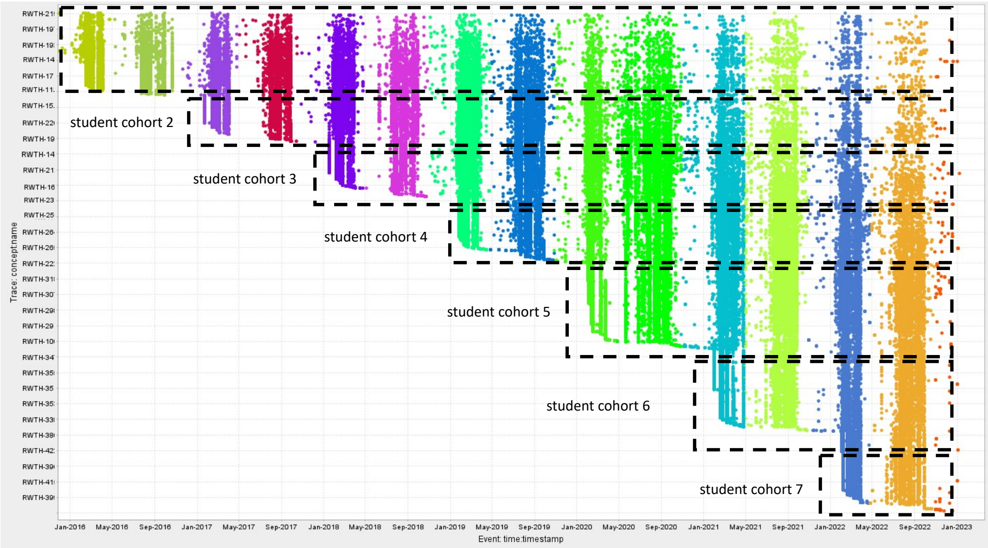


Figure 2: Dotted chart showing exam attempt events of computer science bachelor students from 2016 to 2023. Each row represents a student order by their first exam date. The dotted chart was created using ProM 6.13. The exam attempts are colored by their term.

By assuming the exam date as the timestamp and the student identifier as the case identifier, we get the dotted chart shown in Figure 2. For better readability, the dotted chart in Figure 2 shows only exams from 2016 to 2023. In the chart, each dot represents an event (exam attempt). The dots are aligned horizontally according to the timestamp (exam date) and vertically according to the case (student). Dots on a horizontal line represent a case, i.e., the exam-taking history of a student. The students on the vertical axis are ordered by their first exam date. The color of the dots is defined by the semester the exam belongs to. In the dotted chart, one can see the two exam periods per year. Also, one can see student cohorts of students that start together in the winter semester. For example, student cohort 3 is the class of students that enrolled in the university in 2017 so the first exams they participated in were the exams in the winter semester of 2017 which were written in January 2018. Each cohort of students that enrolled in the same year can be identified in the dotted chart by the semester they participated in their first exams. For each cohort, the number of exam attempts per semester decreases ofer time after their university start because students are dropping out or graduating, especially after six semesters, which is the planned study length. However, for example, for the first cohort, there are still some exam attempts in their 14th semester in September 2022.

There are 1,541 unique courses, but most of them appear very infrequent. For 1,298 courses, less than 20 exam attempts are recorded. The frequency of the 243 courses that appear more than 20 times in the event data are shown in Figure 3. The distribution shows that some courses appear frequently, and most other courses are infrequent.

The average number of attempts over all courses is 1.02 which indicates that most students are able to finish most courses on the first try. On average, a student needs to retake 3.8 courses during his studies. Most of the frequent courses are mandatory courses for the first two semesters. The most frequent course, for example, is the mandatory second-semester course *Linear Algebra* for which 3,665 exam attempts are recorded. The high number of infrequent courses can be explained by students being able to choose from various elective courses and seminars. Especially the seminars are often held only once and are designed for a smaller number of students. Based on these insights, we filtered out infrequent courses that occurred less than 100 times and considered the remaining 73 courses in our analysis.

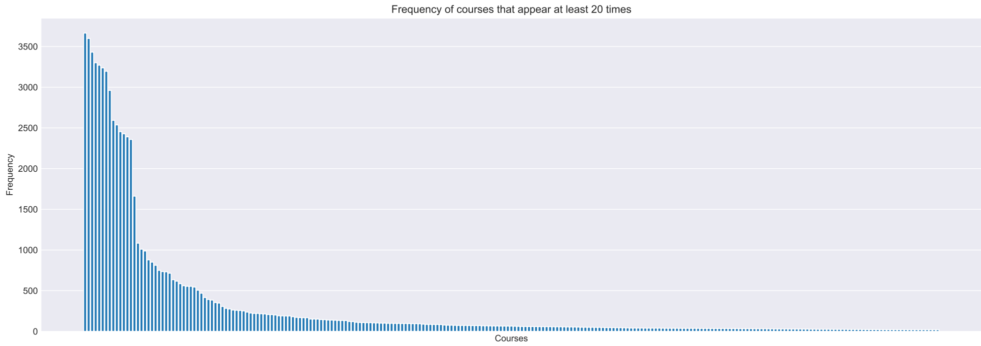


Figure 3: Number of occurrences per course filtered for courses that occur at least 20 times. Resulting in 243 out of 1514 courses.

The grade in an exam is the major performance indicator for a student's success in a course. In the German grading system, grades go from 1.0 (excellent) to 4.0 (passed) and 5.0 (failed). Some courses do not assign grades to the students because they can either be passed or failed. If a course is only pass or fail, the student receives a grade of 0.0 for passing and a grade of 5.0 for failing. We excluded those courses and computed the variance of the grade distribution for the remaining courses. The grade distributions in the form of a boxplot for the ten courses with the highest variance are shown in Figure 4. Most courses with high variance are also more difficult and have a mean grade higher than 3.0. In Figure 5, the grade distributions for the ten courses with the lowest variance are shown in a boxplot. For these courses, the mean grades are better, with half of the courses having a mean grade below 1.5. Out of the 73 courses, 55 have a grade variance higher than 1. Since the range of grades spans only from 1.0 to 5.0, the variance of 1.0 can be considered high. This higher variance implies that these courses pose greater challenges to students, leading to more varied academic results. Consequently, in our evaluation detailed in Section 6, we specifically focus on these courses with a variance above 1.0. This allows us to produce targeted recommendations on what we define as hard courses.

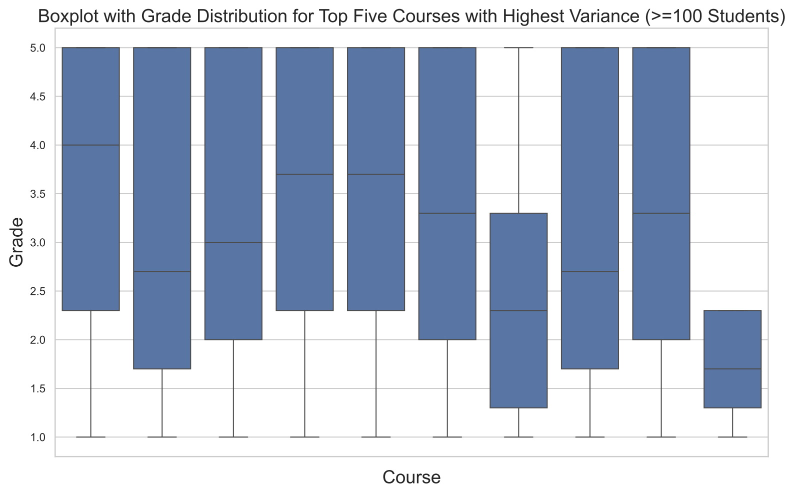


Figure 4: The grade distribution of the ten courses with the highest variance in their grades.

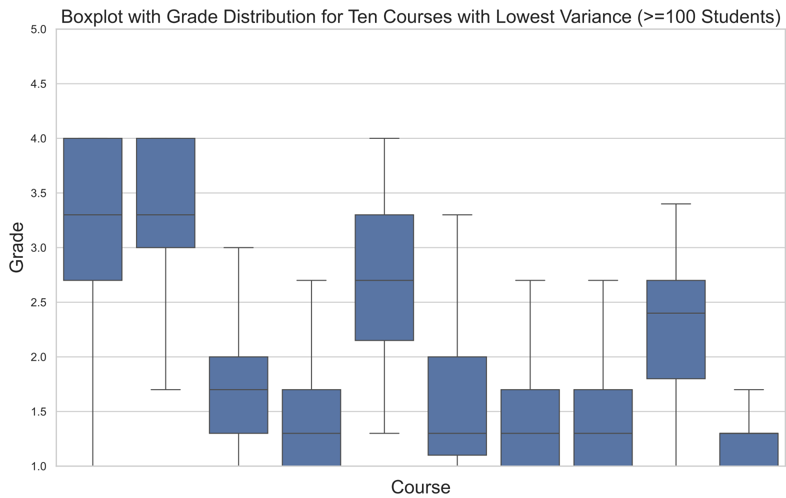


Figure 5: The grade distribution of the ten courses with the lowest variance in their grades.

6 Evaluation

In this section, we assess the effectiveness of our approach, employing the dataset described in Section 5. The evaluation is designed to achieve two primary objectives:

- First, to evaluate the effectiveness of our approach in modeling the relationship between behavioral features and target academic outcomes.

- Second, to determine the practical utility of these models in formulating actionable recommendations that can tangibly enhance academic achievement.

6.1 Evaluating Models

We initiate our evaluation by analyzing the predictive accuracy of the models derived from our approach. For comparison, we adopt decision trees, as introduced in [8], as our baseline model. Note that the selection of the baseline model is carefully aligned with our objectives to ensure not only robust prediction but also high interpretability and direct applicability of the results. We acknowledge that while other models like logistic regression and random forests are valuable in predictive analytics, they do not meet the crucial criterion of generating interpretable, actionable rules required for direct application in educational program design.

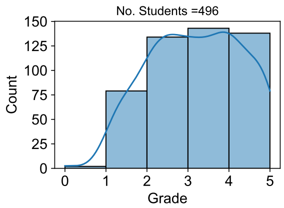
We train each model to predict the GPA up to N-th semester (cf. Subsection 4.1). To circumvent data leakage and ensure fairness in model assessment, we exclude any features directly tied to GPA, focusing instead on the other informative features outlined in Subsection 4.2. We use features such as occurrences, first occurrences of a course, last occurrence of a course, total credits, total credits in a semester, study duration, and course duration.

We segment the event data detailed in Section 5 into five distinct datasets corresponding to different academic stages. For instance, *Data-S2* encompasses events for students concluding their second semester and embarking on their third. Each dataset is used to train models to predict the GPA up to N-th semester. For example, we use *Data-S2* to train a model to predict the GPA up to 2nd semester. This segmentation enables an assessment of model performance across varying academic phases, enriching our understanding of the models' predictive capabilities at different points in a student's academic journey. The distribution of overall GPA within each dataset, illustrated in Figure 6, offers insights into the academic progression captured in our data.

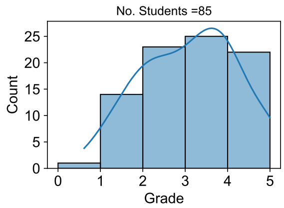
To ensure the reliability and generalizability of our results, we employ a 5-fold cross-validation technique. This method divides each dataset into five subsets, ensuring a comprehensive and robust evaluation by iteratively training and validating the models on diverse data splits, thereby mitigating the risk of overfitting.

Model performance is evaluated using Mean Squared Error (MSE), which calculates the average of the squares of the errors between actual and predicted GPAs. Figure 7 illustrates the comparative performance of our method against a baseline model, highlighting the accuracy of GPA predictions through cross-validation folds. Notably, our approach consistently outperforms the baseline, evidencing its superior predictive capability.

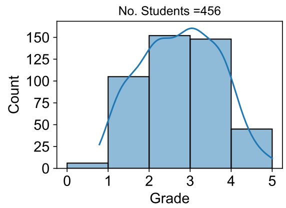
Particularly in Data-S3, where a decision tree model's MSE exceeded 2, our approach demonstrated remarkable robustness. Nonetheless, there is a notable variance in model performance in Data-S2 and Data-S5. The second semester's data likely lack sufficient behavioral indicators, as students' patterns are not yet well-established. In the fifth semester, the smaller student cohort might limit the model's ability to generalize effectively, potentially impacting its performance.



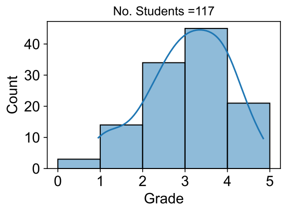
(a) Data-S2



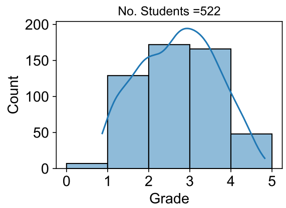
(b) Data-S3



(c) Data-S4



(d) Data-S5



(e) Data-S6

Figure 6: Distribution of overall GPA within each dataset, illustrating academic progression.

6.2 Effectiveness of Recommendation Generation

Beyond predictive accuracy, the utility of our approach extends to its capability to generate meaningful recommendations to enhance student success. This subsection assesses the recommendation quality, focusing on the interpretability and applicability of the generated recommendations.

Before generating recommendations, we report the number of rules generated per semester, for each of which we can generate a recommendation, in Figure 8. The chart highlights the fluctuation in rule quantity and, subsequently, recommendation quantity across different academic periods. Notably, the fifth semester stands out with the highest rule count at 143, while the third semester presents the minimum at 65 rules, highlighting that the students in the fifth semester show more distinctive patterns that lead to higher GPA. The chart further distinguishes between positive rules, which, if adhered to, could boost academic performance, and negative rules, avoidance of which might similarly result in performance gains. This differentiation reveals diverse trends across semesters, with a general inclination towards generating more positive than negative rules in most semesters. An exception is observed in the fifth semester, where the number of negative rules surpasses the positive ones.

Next, by interpreting the extracted rules, we derive actionable recommendations. Table 3 showcases the top 10 rules extracted from the predictive model targeting the overall GPA prediction for students in their fourth semester. The rules are initially prioritized by their importance measures. Subsequently, the rules are evaluated based on their coefficients, which indicate the direction and magnitude of their impact on GPA. For instance, the first rule, “last($\texttt{SAP}$) ≤ 2.5 & total_credits > 77.0”, suggests a

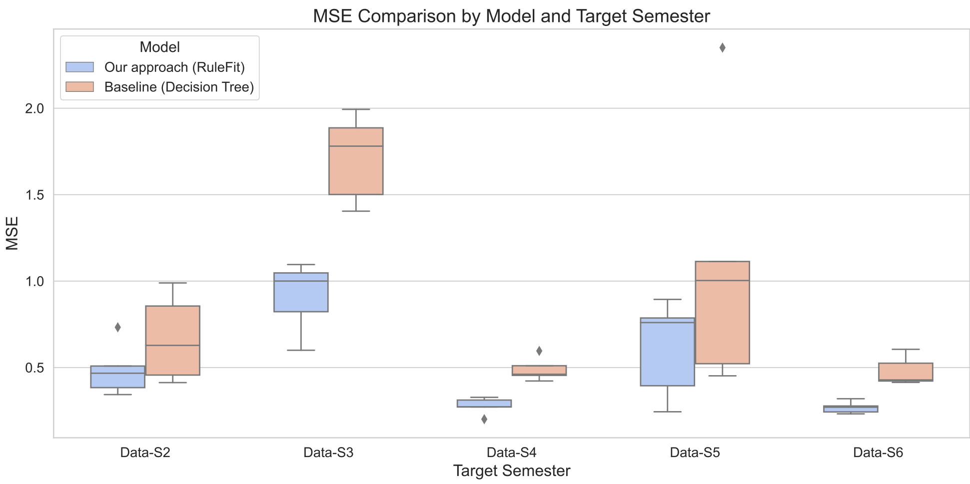


Figure 7: Comparison of predictive performance between the proposed approach and decision tree baseline, measured in terms of Mean Squared Error (MSE) across 5-fold cross-validation.

significant enhancement in GPA (considering our system where a lower GPA is more favorable) when students complete `SAP` before their third semester and accumulate a total credits higher than 77.

Rules	Importance	Coefficient
last(`SAP`) $\leq$ 2.5 & total_credits > 77.0	0.0898	-0.1799
last(`DA`) $\leq$ 4.5 & last(`AI`) $\leq$ 0.5 & course_duration(`SAP`) $\leq$ 1	0.0805	-0.176
last(`SAP`) $\leq$ 8.0 & last(`OR`) $\leq$ 0.5 & last(`Wt`) $\leq$ 0.5 & total_courses $\leq$ 16.5	0.067	-0.1363
total_credits $\leq$ 90.5	0.063	0.1371
course_duration(`BS`) $\leq$ 1 & total_credits > 7 & total_courses $\leq$ 15.5	0.0622	-0.1276
total_credits $\leq$ 39 & last(`St`) $\leq$ 5	0.0609	0.1531
first(`DS`) $\leq$ 1.5 & total_courses > 15.5 & course_duration(`SAP`) $\leq$ 1 & total_credits > 41.5 & last(`SAP`) > 0.5	0.0586	-0.1573
total_course > 15.5 & total_credits $\leq$ 100.5 & total_credits > 93.5	0.0584	0.2521
first(`EW`) $\leq$ 1 & last(`KSP`) $\leq$ 1 & total_credits > 96.5	0.0541	-0.136
total_credits > 61.5 & course_duration(`ETI`) $\leq$ 1 & total_credits $\leq$ 96.5 & course_duration(`BS`) $\leq$ 1 & total_courses $\leq$ 14.5	0.0529	-0.1229

Table 3: Top 10 rules extracted from the predictive model for fourth-semester GPA prediction

By interpreting these rules within the educational context, we can provide targeted recommendations to course designers, such as focusing on completing `SAP` in the first half of the study and managing students' course load to less than 77 credits.

Furthermore, Table 4 shows the top 10 rules obtained from the predictive model targeting the overall GPA for students in their fifth semester. For instance, the rule

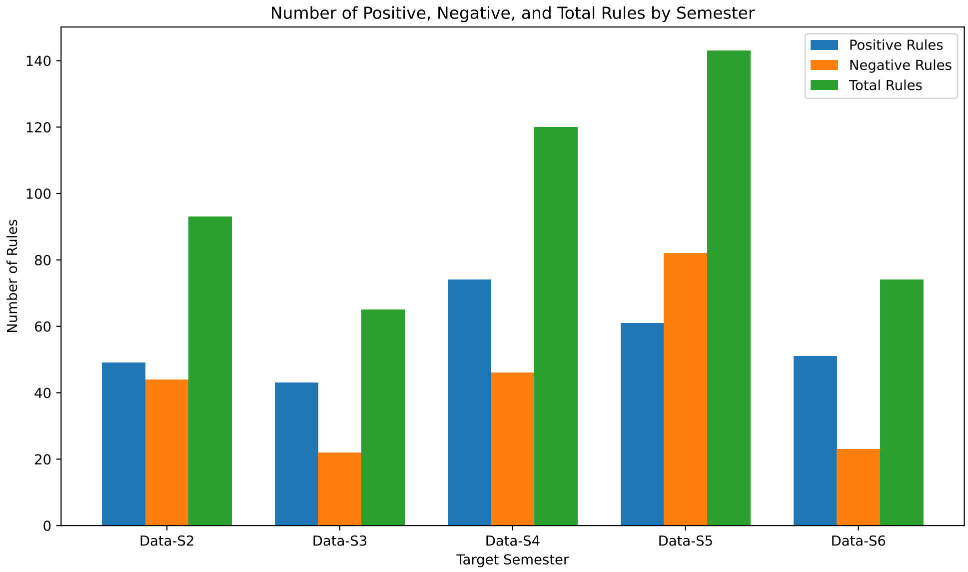


Figure 8: The number of rules extracted from predictive models for different target semesters.

"first(`ST`) $\leq$ 4.5 & last(`NA`) $\leq$ 1.0 & last(`Comp`) $\leq$ 1.5" with a coefficient of 0.4392 indicates that enrolling in `ST` (mandatory course) later in the academic journey while enrolling in "`NA`" and "`Comp`" (electives) early on is associated with an unfavorable increase in GPA.

Rules	**Importance**	**Coefficient**
first(`ST`) $\leq$ 4.5 & last(`NA`) $\leq$ 1.0 & last(`Comp`) $\leq$ 1.5	0.2131	0.4392
course_duration(`LA`) $\leq$ 1.0 & course_duration(`OR`) $\leq$ 1.0 & course_duration(`Prog`) $\leq$ 1.0 & course_duration(`TI`) $\leq$ 1.0 & total_courses $\leq$ 17.5	0.1405	-0.2853
first(`BWL`) $\leq$ 0.5	0.1278	0.3922
last(`DuS`) $\leq$ 6 & course_duration(`BS`) $\leq$ 3 & total_credits $\leq$ 99 & first(`LA`) $\leq$ 6 & last(`DS`) $\leq$ 0.5 & total_courses $\leq$ 17	0.1132	-0.2298
last(`ETI`) $\leq$ 0.5 & last(`BS`) $\leq$ 0.5 & last(EaS) $\leq$ 2.5	0.111	-0.4382
total_credits $\leq$ 101.5 & last(`LA`) $\leq$ 1.5	0.1103	0.2227
total_credits $>$ 32.5 & last(`MIT`) $\leq$ 0.5 & first(`Sp`) $>$ 2.5	0.1099	0.2375
total_courses $>$ 11.5 & first(`Sp`) $\leq$ 0.5	0.1094	0.3022
total_credits $>$ 17.5 & last(`Mg`) $\leq$ 1 & first(`IAI`) $\leq$ 2.5 & last(`AI`) $\leq$ 1.5	0.1041	-0.2083
total_credits $>$ 46 & first(`Cp`) $\leq$ 1.5 & course_duration(`SAP`) $\leq$ 1 & last(`DaS`) $\leq$ 6.5 & total_credits $\leq$ 107.5	0.095	-0.191

Table 4: Top 10 rules extracted from the predictive model for fifth-semester GPA prediction

Conversely, the rule with a negative coefficient, "course_duration(LA) $\leq$ 1.0 & course_duration(OR) $\leq$ 1.0 & course_duration(Prog) $\leq$ 1.0 & course_duration(TI) $\leq$ 1.0 & total_courses $\leq$ 17.5", suggests that timely completion of these foundational courses, alongside managing a moderate total course load, positively impacts the GPA.

Using these insights, we can develop targeted recommendations for study program designers. The emphasis could be on advising the optimal sequencing of courses and highlighting the importance of foundational subjects like ST in the early semesters. In addition, recommendations might focus on strategic planning of students' academic trajectory, ensuring that critical courses, such as LA, OR, Prog, and TI, are not only selected but also completed in a timely manner to leverage their positive impact on GPA.

While our model inherently reflects the structured progression pathways mandated by existing academic regulations, it also allows for the exploration of atypical yet successful academic behaviors, e.g., last(EaS) $\leq$ 2.5 although the course is recommended for the fourth semester. This capability can highlight opportunities for program designers to re-evaluate their course sequencing and prerequisite structures. Such insights are particularly valuable in evolving academic environments where flexibility can significantly enhance student outcomes.

7 Conclusion

In this paper, we introduced a novel approach for generating actionable recommendations from educational event data, specifically those collected by CMS. Our approach, structured in three distinct phases, leverages this data to uncover insights into students' academic behaviors and their implications on key educational outcomes. First, we identified essential features related to students' academic performance, examining aspects such as grade trends, course sequencing, and exam attempt patterns. To that end, we adapted the features suggested by de Leoni et al. [9] to suit the educational context, ensuring their relevance. The core of our approach is the deployment of the RuleFit algorithm to train a predictive model capable of forecasting crucial educational outcomes like GPAs. This model not only serves as a predictive tool but also as a means to distill meaningful rules that underpin these forecasts. Finally, we convert these rules into tangible recommendations for study program designers.

Our approach was evaluated using a dataset from the computer science bachelor's program at RWTH Aachen University. By extracting features for a cohort of students, developing a model to predict their overall GPA, and subsequently deriving recommendations that lead to higher GPAs, we showcased the potential of our approach to inform and enhance educational practices.

Our proposed approach still has several limitations. First, the approach is evaluated in a specific program at RWTH Aachen University. This raises questions about the method's applicability to other disciplines, educational levels, or institutions with different academic structures and cultures. The uniqueness of each educational setting and student demographics may require adjustments in the proposed approach. Second, educational practices, course content, and program requirements evolve over time. This dynamic nature of the educational environment could render the recommendations derived from historical data less applicable or even obsolete, necessitating continual model updates and recalibrations to align with current educational practices. Third, the model primarily focuses on observable academic behaviors and course-related data, potentially overlooking the personal and socio-economic factors that influence student performance. Factors such as personal motivation and varying learning styles can significantly impact academic outcomes but are often not captured in CMS event data. Fourth, the lack of a structured mechanism for feedback in the model's development limits the opportunity for continuous improvement and adaptation to users' needs. Engaging students and educators in providing feedback on the utility, accuracy, and impact of the recommendations could provide insights into the model's real-world effectiveness. Finally, while our approach does not incorporate instructor-specific features due to the complexities of privacy and data protection, we recognize the potential insights such data could provide. Future research could explore the impact of varying instructional methods on student outcomes.

Acknowledgements. The authors gratefully acknowledge the financial support by the Federal Ministry of Education and Research (BMBF) for the joint project AIStudyBuddy (grant no. 16DHBKI016).

SPONSORED BY THE

Federal Ministry
of Education
and Research

Declarations

Competing Interests

The authors declare no competing interests.

Ethics Approval

Ethics approval was not required for this research.

Consent to Participate

No one participated in the study of the manuscript.

Data Availability

The research data supporting the findings of this manuscript are based on educational event data sourced from Campus Management Systems, which contains sensitive and private information related to students. Due to privacy concerns and the data's ownership by RWTH Aachen University, the dataset cannot be made openly available.

Researchers interested in accessing the data for academic purposes may submit a formal request to RWTH Aachen University. Each request will be evaluated on a case-by-case basis, considering ethical guidelines and data protection regulations. Access may be granted under strict conditions that ensure the privacy and confidentiality of the individuals represented in the dataset.

Author Contribution

All authors conceptualized the approach proposed in the manuscript. G.P. wrote the main manuscript text. L.L. wrote the data description and scoping. W.v.d.A supervised and validated the research project. All authors reviewed the manuscript.

References

[1] Catherine Bovill, C.J.B., Morss, K.: Engaging and empowering first-year students through curriculum design: perspectives from the literature. Teaching in Higher Education **16**(2), 197–209 (2011) https://doi.org/10.1080/13562517.2010.515024 https://doi.org/10.1080/13562517.2010.515024

[2] Al-Twijri, M.I., Luna, J.M., Herrera, F., Ventura, S.: Course recommendation based on sequences: An evolutionary search of emerging sequential patterns. Cogn. Comput. **14**(4), 1474–1495 (2022) https://doi.org/10.1007/S12559-022-10015-5

[3] Santos, G.A.S., Belloze, K.T., Tarrataca, L., Haddad, D.B., Bordignon, A.L., Brandão, D.N.: Evolvedtree: Analyzing student dropout in universities. In: 2020 International Conference on Systems, Signals and Image Processing, IWSSIP 2020, Niterói, Brazil, July 1-3, 2020, pp. 173–178 (2020). https://doi.org/10.1109/IWSSIP48289.2020.9145203

[4] Robles, F.B., Köhler, J., Hinrechsen, K., Araya, V., Hidalgo, L., Jara, J.L.: Using machine learning methods to identify significant variables for the prediction of first-year informatics engineering students dropout. In: 39th International Conference of the Chilean Computer Science Society, SCCC 2020, Coquimbo, Chile, November 16-20, 2020, pp. 1–5 (2020). https://doi.org/10.1109/SCCC51225.2020.9281280

[5] Yagci, M.: Educational data mining: prediction of students' academic performance using machine learning algorithms. Smart Learn. Environ. **9**(1), 11 (2022) https://doi.org/10.1186/S40561-022-00192-Z

[6] Veerasamy, A.K., D'Souza, D.J., Apiola, M., Laakso, M., Salakoski, T.: Using early assessment performance as early warning signs to identify at-risk students in programming courses. In: IEEE Frontiers in Education Conference, FIE 2020, Uppsala, Sweden, October 21-24, 2020, pp. 1–9 (2020). https://doi.org/10.1109/FIE44824.2020.9274277

[7] Alangari, N., Alturki, R.: Predicting students final gpa using 15 classification algorithms. Romanian Journal of Information Science and Technology **23**(3), 238–249 (2020)

[8] Rafiei, M., Bayrak, D., Pourbafrani, M., Park, G., Helal, H., Lakemeyer, G., van der Aalst, W.M.P.: Extracting rules from event data for study planning. CoRR

abs/2310.02735 (2023) https://doi.org/10.48550/ARXIV.2310.02735

[9] de Leoni, M., van der Aalst, W.M.P., Dees, M.: A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. Inf. Syst. **56**, 235–257 (2016) https://doi.org/10.1016/J.IS.2015.07.003

[10] Friedman, J.H., Popescu, B.E.: Predictive learning via rule ensembles. The Annals of Applied Statistics **2**(3), 916–954 (2008)

[11] York, T.T., Gibson, C., Rankin, S.: Defining and measuring academic success. Practical assessment, research, and evaluation **20**(1), 5 (2019)

[12] Vella, E.J., Turesky, E.F., Hebert, J.: Predictors of academic success in web-based courses: Age, gpa, and instruction mode. Quality Assurance in Education **24**(4), 586–600 (2016)

[13] Al-Barrak, M.A., Al-Razgan, M.: Predicting students final gpa using decision trees: a case study. International journal of information and education technology **6**(7), 528 (2016)

[14] Rizvi, S., Rienties, B., Khoja, S.A.: The role of demographics in online learning; a decision tree based approach. Computers & Education **137**, 32–47 (2019)

[15] Deasy, C., Coughlan, B., Pironom, J., Jourdan, D., Mannix-McNamara, P.: Psychological distress and coping amongst higher education students: A mixed method enquiry. Plos one **9**(12), 115193 (2014)

[16] Puri, G.: Critical success factors in e-learning–an empirical study. International Journal of Multidisciplinary Research **2**(1), 149–161 (2012)

[17] Bogarín, A., Cerezo, R., Romero, C.: A survey on educational process mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **8** (2017)

[18] Trcka, N., Pechenizkiy, M., van der Aalst, W.M.P.: Process mining from educational data. Handbook of educational data mining, 123–142 (2010)

[19] Salazar-Fernandez, J.P., Sepúlveda, M., Munoz-Gama, J.: Influence of student diversity on educational trajectories in engineering high-failure rate courses that lead to late dropout. In: 2019 IEEE Global Engineering Education Conference (EDUCON), pp. 607–616 (2019). https://doi.org/10.1109/EDUCON.2019.8725143

[20] Zhang, L., Liu, X., Liu, X.: Personalized instructing recommendation system based on web mining. In: Proceedings of the 9th International Conference for Young Computer Scientists, ICYCS 2008, Zhang Jia Jie, Hunan, China, November 18-21, 2008, pp. 2517–2521 (2008). https://doi.org/10.1109/ICYCS.2008.115

[21] Zaïane, O.R.: Building a recommender agent for e-learning systems. In: International Conference on Computers in Education, ICCE 2002, Auckland, New Zealand, December 3-6, 2002, Volume 1, pp. 55–59 (2002). https://doi.org/10.1109/CIE.2002.1185862

[22] Wang, F.-H.: Content recommendation based on education-contextualized browsing events for web-based personalized learning. Journal of Educational Technology & Society **11**(4), 94–112 (2008)

[23] Wang, F., Shao, H.: Effective personalized recommendation based on time-framed navigation clustering and association mining. Expert Syst. Appl. **27**(3), 365–377 (2004) https://doi.org/10.1016/J.ESWA.2004.05.005

[24] Su, Z., Song, W., Lin, M., Li, J.: Web text clustering for personalized e-learning based on maximal frequent itemsets. In: 2008 International Conference on Computer Science and Software Engineering, vol. 6, pp. 452–455 (2008). https://doi.org/10.1109/CSSE.2008.1639

[25] Bentéjac, C., Csörgo, A., Martínez-Muñoz, G.: A comparative analysis of gradient boosting algorithms. Artif. Intell. Rev. **54**(3), 1937–1967 (2021) https://doi.org/10.1007/S10462-020-09896-5

[26] Tibshirani, R.: Regression Shrinkage and Selection Via the Lasso. Journal of the Royal Statistical Society: Series B (Methodological) **58**(1), 267–288 (2018) https://doi.org/10.1111/j.2517-6161.1996.tb02080.x