12-2023

# Analysis of Student Behavior and Score Prediction in Assistments Online Learning

Aswani Yaramala
*Utah State University*, aswani.yaramala@usu.edu

ANALYSIS OF STUDENT BEHAVIOR AND SCORE PREDICTION IN

ASSISTMENTS ONLINE LEARNING

by

Aswani Yaramala

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Computer Science

Approved:

| | |
|---|---|
| Hamid Karimi, Ph.D.<br>Major Professor | John Edwards, Ph.D.<br>Committee Member |
| Shuhan Yuan, Ph.D.<br>Committee Member | D. Richard Cutler, Ph.D.<br>Vice Provost of Graduate Studies |

UTAH STATE UNIVERSITY
Logan, Utah

2023

# ABSTRACT

Analysis of Student Behavior and Score Prediction in ASSISTments Online Learning

by

Aswani Yaramala, MASTER OF SCIENCE

Utah State University, 2023

Major Professor: Hamid Karimi, Ph.D.
Department: Computer Science

This thesis presents an in-depth analysis of student behavior and score prediction in the ASSISTments online learning platform. Leveraging the comprehensive dataset from the EDM Cup 2023 Kaggle Competition, we address four research questions related to the impact of tutoring materials, skill mastery, feature extraction, and graph representation learning. To investigate the impact of tutoring materials, we analyze the influence of students requesting hints and explanations on their performance in end-of-unit assignments. Our findings provide insights into the role of guidance in learning and inform the development of superior tutoring strategies. Additionally, we explore the correlation between mastery/non-mastery of specific skills during in-unit problems and performance in corresponding end-of-unit assignments, shedding light on the efficacy of standard-aligned curricula. In terms of feature extraction, we extract relevant features from extensive student activity data and determine their importance in predicting assignment grades. This enhances student performance prediction, aiding the early identification of at-risk students and enabling effective monitoring of progress. Furthermore, we employ graph representation learning techniques to model the complex relationships between different entities in the dataset. This yields a more nuanced understanding of factors influencing student performance and facilitates the development of more accurate predictive models. Overall, our study contributes to the theoretical understanding and practical application of data mining techniques in online learning contexts, with implications for personalized learning, interventions, and support mechanism. Our code is publicly available in https://github.com/DSAatUSU/EDMCup2023.

(77 pages)

# PUBLIC ABSTRACT

Analysis of Student Behavior and Score Prediction in ASSISTments Online Learning

Aswani Yaramala

Understanding and analyzing student behavior is paramount in enhancing online learning, and this thesis delves into the subject by presenting an in-depth analysis of student behavior and score prediction in the ASSISTments online learning platform. We used data from the EDM Cup 2023 Kaggle Competition to answer four key questions. First, we explored how students seeking hints and explanations affect their performance in assignments, shedding light on the role of guidance in learning. Second, we looked at the connection between students mastering specific skills and their performance in related assignments, giving insights into the effectiveness of curriculum alignment. Third, we identified important features from student activity data to improve grade prediction, helping identify at-risk students early and monitor their progress. Lastly, we used graph representation learning to understand complex relationships in the data, leading to more accurate predictive models. This research enhances our understanding of data mining in online learning, with implications for personalized learning and support mechanisms.

To my lovely husband and adorable kiddo

# ACKNOWLEDGMENTS

I want to extend my sincere appreciation to Dr. Hamid Karimi, my advisor, for his consistent guidance and support throughout this journey. I am truly grateful for his unwavering encouragement and understanding as a mentor, especially for someone like me who is relatively new to this field. His ability to attentively listen, provide assistance, and address all my inquiries has been invaluable.

I would also like to express my gratitude to Dr. John Edwards and Dr. Shuhan Yuan for their contributions to my graduate committee and for offering helpful suggestions during my time at Utah State University. Special thanks go to my dear friend Soheila Farokhi for her priceless guidance and support. I extend my appreciation to all the members of the DSA lab at USU and acknowledge the support received from the CS department at USU.

I am profoundly grateful to my parents, Thulasi and Niranjan, as well as my brother, Pavan, whose encouragement has played a pivotal role in bringing me to the position I am in today. I extend my appreciation to my in-laws for their support. I also want to express my gratitude to my husband, Mohan, who has been an incredible co-parent and partner in this journey. His understanding and encouragement have enabled me to balance the demands of academia and family life. A special note of thanks to my three-year-old son, Nihant, whose boundless energy and infectious laughter brought joy to the challenging moments of this academic pursuit.

Finally, I am grateful to all my friends, particularly Sriya and Nitya, for making this journey more memorable.

Aswani Yaramala

CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

Emerging technologies and evolving societal needs have propelled the digital transformation in education. A significant catalyst in this digital revolution is the rapid rise of online learning platforms, which has seen an even more profound upsurge amid the global shift towards remote learning necessitated by recent events [1, 2]. These platforms serve to democratize education by breaking down geographical barriers and making quality education accessible to individuals irrespective of their location [3]. In addition, they extend the reach of education to under-served communities and individuals with restricted mobility, ensuring that everyone has the opportunity to learn and grow [4]. Furthermore, online learning platforms generate a wealth of data ranging from detailed clickstream data to more structured assignment and assessment data. This data-rich environment presents an unprecedented opportunity to gain insights into the learning process and student behavior [5]. Leveraging data mining and machine learning techniques, this wealth of data can be harnessed to predict student performance, provide personalized learning experiences, and ultimately enhance learning outcomes [6]. The potential of such a comprehensive, data-driven approach lies in its capacity to shape a responsive, student-centric educational landscape that adapts to the unique needs and progress of each learner.

Two particular areas of interest in this data-rich landscape are analyzing student behavior and predicting exam assignment grades. Student behavior analysis in online learning environments plays a pivotal role in understanding the learning process. With the ability to track and record every interaction of students within the learning platform, we can uncover intricate patterns that characterize different learning behaviors. Detailed analysis of these behaviors can lead to valuable insights into students' engagement, motivation, learning strategies, and potential difficulties [7, 8]. This information, in turn, can inform the design of personalized learning paths, interventions, and support mechanisms, contributing to improved learning outcomes and overall student success. Parallel to this, predicting exam assignment grades is another critical aspect of educational data mining. Predictive models can leverage student behavior data and other relevant features to forecast their performance in future assessments. This capability is not only essential for the timely identification of students at risk of underperforming or dropping out but also valuable for students and educators in monitoring progress and adjusting learning or teaching strategies accordingly [7, 8].

With the above discussion, in this thesis, we present analyses, developed machine learning models, and experimental results while participating in the EDM CUP 2023 competition. In the competition under discussion, the primary challenge was predicting students' scores on end-of-unit assignments, utilizing the clickstream data from all the in-unit assignments the students had previously completed. The comprehensive dataset for this project was not only constituted by student behaviors but also included additional information about the curricula, individual assignments, problem statements, and the tutoring support provided to the students, all of which were utilized to enhance the accuracy of our predictive models. Figure 1.1 illustrates an overview of the analytical framework for student behavior and score prediction used in this study. In the context of this competition, we aim to answer the following research questions. We also describe the significance of each research question as to how it would contribute to the existing body of knowledge.

- **RQ1.** What is the impact of students requesting tutoring materials (e.g., hints and explanations) on their performance in end-of-unit assignments?

  - **Significance.** Investigating the impacts of requesting tutoring materials allows for insights into the role of guidance in learning and how student initiative to seek help influences outcomes. Furthermore, the findings can inform the development of superior tutoring strategies, potentially improving learning experiences. To answer this question, we employ quasi-experiments to estimate the causal impacts of the request for tutoring materials on end-of-unit assignments.

- **RQ2.** What patterns exist in the correlation between mastery of specific skills during in-unit problems and performance in corresponding end-of-unit assignments?

  – **Significance.** This question aims to establish a connection between the mastery of specific skills, aligned with the Common Core State Standards (CCSS) for mathematics [9], during in-unit assignments and performance in corresponding end-of-unit assignments. By analyzing how progress in CCSS-aligned skills during unit tasks influences overall performance on end-of-unit assignments, we can better understand how effective standard-aligned curricula foster student mastery. Furthermore, the insights derived from this question can shape the design of targeted interventions and support mechanisms, possibly leading to improved learning experiences and alignment with CCSS guidelines. To answer this question, we arrange previously completed problems and end-of-unit problems as a set of items related to a student. We then employ association rule mining [10,11] to extract dominant rules showing how previous skills (rule antecedents) influence the end-of-unit assignment skill (rule consequent).

- **RQ3.** How can relevant features be extracted from the extensive student activity data within the ASSISTments online learning platform for use in a predictive machine learning model for end-of-unit assignment grades? Also, what is the relative importance of the extracted features in predicting assignment grades?

  – **Significance.** Feature extraction and understanding their importance in grade prediction are critical for developing more accurate predictive models [12–14]. Enhancing student performance prediction allows for earlier identification of students at risk of under-performance [15], enabling timely support. Furthermore, it aids students and educators in effectively monitoring progress and adjusting strategies as necessary. We carry out extensive feature extraction and engineering from various dataset attributes to address this question. We then use these features as input to traditional, robust machine learning algorithms like Random Forest. Finally, we run thorough experiments and determine each feature's importance.

- **RQ4.** How can graph representation learning be employed to model the complex relationships between different entities in the dataset (e.g., teacher, students, problems)? And do the features extracted from these relationships yield any predictive power in predicting assignment grades?

  – **Significance.** Leveraging graph representation learning to model complex dataset relationships can lead to a more nuanced understanding of the factors influencing student performance [15,16]. It could result in developing more sophisticated predictive models, thereby improving grade prediction accuracy and educational outcomes by enabling more precise and timely interventions. To address this question, we first represent the complex relationship between different entities in the dataset as a graph. We then apply a graph representation learning algorithm called node2vec [17] to extract salient features from the underlying graph. It is important to note that graph representation learning uses machine learning techniques to capture and encode the properties, features, and structures of graphs into vector representations, aiding the understanding and analysis of complex relational data [18].

Overall, these research questions are significant as they aim to explore and understand different aspects of student engagement, behavior, and learning patterns in an online learning context, specifically within the ASSISTments platform. In summary, our contributions in this study are as follows:

① We comprehensively investigate the influence of requested tutoring materials on student performance in an online learning context. In addition, our quasi-experimental approach offers valuable insights into the causal effects of these elements on end-of-unit assignments.

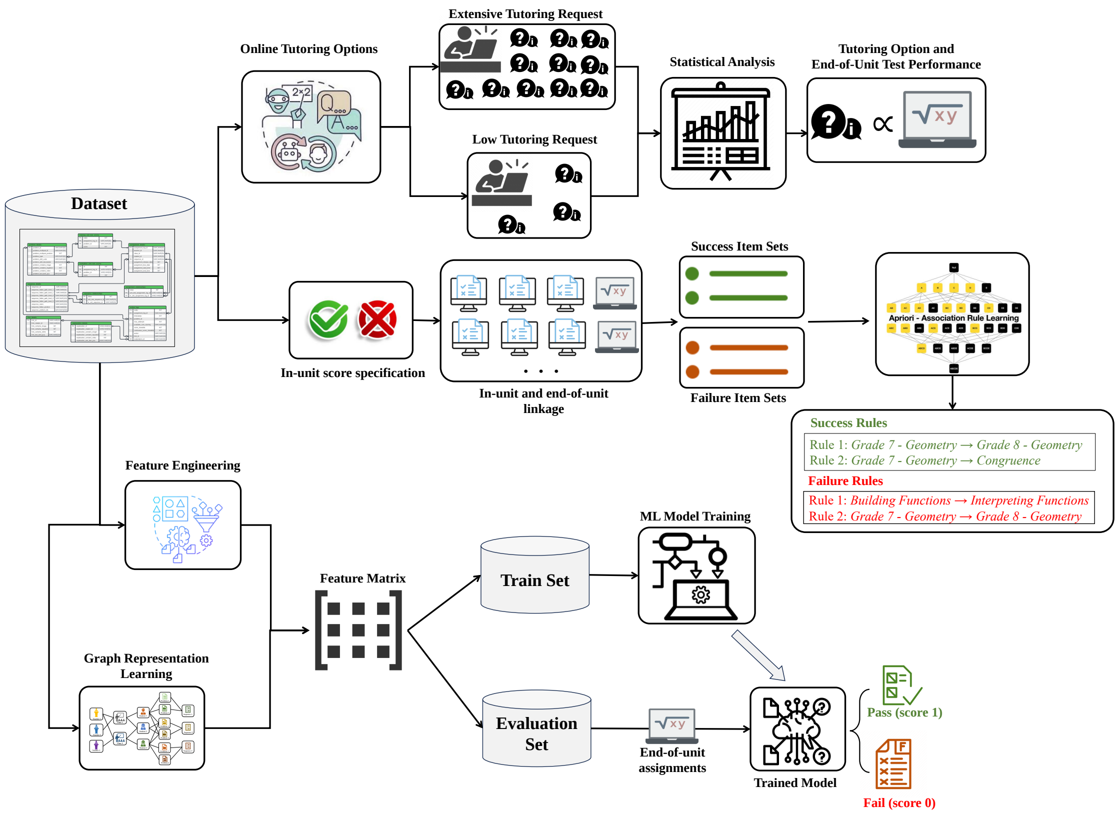


Fig. 1.1: Framework for student behavior analysis and score prediction, used in this study, divided into three primary sections. The top section shows our investigation into the link between tutoring requests and student performance on end-of-unit tests. The middle portion of the figure outlines our methodology for unearthing informative rules that elucidate the mastery or non-mastery of Common Core State Standards (CCSS) skills during in-unit tests linked to end-of-unit skill mastery/non-mastery and general student performance. Lastly, the bottom segment details the employment of graph representation learning and machine learning predictive modeling to forecast student scores on end-of-unit tests.

② We establish a clear connection between the mastery of specific Common Core State Standards (CCSS)-aligned skills during in-unit problems and performance on end-of-unit assignments. This contributes to a better understanding of the efficacy of standard-aligned curricula.

③ We conduct extensive feature extraction from various dataset attributes and establish their importance in grade prediction tasks. Our approach enables a better prediction of student performance, aiding the early detection of at-risk students.

④ We apply graph representation learning algorithms to model complex relationships in the dataset. This novel application provides a more nuanced understanding of the factors influencing student performance.

⑤ Through our comprehensive approach, we not only answer pertinent research questions in the field of online learning but also provide a methodological framework that can be replicated in other similar studies. This can enhance the field of educational data mining.

⑥ Our analytical framework distinguished itself by securing the fifth position on both the public and private leaderboards amidst competition from 49 teams[1]. Our public score, in terms of Area Under the Curve (AUC), stood at 0.78072, merely 0.01068 points behind the leading team. Similarly, our private score of 0.78579 was close behind the top team, with a minimal difference of 0.0039. This result highlights the efficacy of our approach.

The organization of this thesis is as follows. Chapter 2 starts with a review of the existing literature pertinent to our study. Next, Chapter 3 presents a description of the dataset used in our analysis as well as an initial data exploration. In Chapter 4, we delve into an in-depth analysis of the data, providing answers to **RQ1** and **RQ2** related to the effects of tutoring materials and skill mastery on student performance. Subsequently, Chapter 5 focuses on the process of feature extraction, the application of graph representation

[1] https://www.kaggle.com/competitions/edm-cup-2023/leaderboard

learning, the design of predictive models, the execution of classification experiments, and the discussion of their results, addressing **RQ3** and **RQ4**. Finally, we conclude our study in Chapter 6, summarizing our key findings, acknowledging the limitations of our analytical framework and the dataset, and contemplating potential directions for future research.

# CHAPTER 2

# RELATED WORK

The rapid expansion of online learning platforms and the accompanying surge in data collection have triggered a burgeoning area of research in Educational Data Mining (EDM) and Learning Analytics (LA). The synergy of EDM and LA has manifested a profound capacity to decipher and employ educational data for enhancing learning and teaching methods. These fields intricately interlace to parse educational environments and tailor pedagogical strategies to individual learning needs [15,19–22]. The assimilation of sophisticated machine learning techniques in EDM and LA has not only refined predictive models for student success but has also provided a granular understanding of educational interactions within online platforms [19].

Existing literature related to our study can be broadly categorized into four main themes: (1) The role of tutoring materials in online learning, (2) the correlation between in-unit skill mastery and end-of-unit performance, (3) feature extraction and importance in predictive modeling, and (4) the use of graph representation learning in predicting student performance.

## 2.1 Tutoring Materials in Online Learning

Intelligent Tutoring Systems (ITS) have sparked significant interest among researchers and educators over several decades. Various studies and publications have explored the potential and challenges of ITS, particularly in collaborative learning. In [23], the researchers delved into expanding the Cognitive Tutor framework to facilitate the development of ITS for collaborative learning. The study also reflected on the efficacy of the improved authoring tools and their role in this context. Similarly, [24] provided a comprehensive overview of ITS, encompassing the design, development, and incorporation of ITS for collaborative learning. It expanded on subjects like group modeling, social learning, and the application of ITS in online learning environments. The paper [25] is another crucial work in this field. The authors discussed the development, execution, and lessons derived from implementing cognitive tutors in varied educational contexts. These tutors are ITS that leverage cognitive models to offer personalized instruction to students. Moreover, [26] presented the potential of AI systems, including AI tutoring systems and AI teaching assistants, in augmenting learner-instructor interaction within online learning scenarios. The authors also introduced a theoretical framework for examining this impact. Another important research [27] analyzed the challenges mathematics teachers face when utilizing a digital mathematics textbook integrated with an ITS. The paper discussed the ITS features and results from an effectiveness study. On a related note, [28] conducted an empirical analysis of student behavior in asynchronous online courses. This paper utilized data mining techniques to discern patterns in student behavior and to forecast student achievement. Also, [29] explored the influence of assistance on learning and emotional states during problem-solving activities with a computer tutor. This research compared the effectiveness of various assistance types, such as worked examples and hints, on student learning and effect. Lastly, the proceeding “Intelligent Tutoring Systems Conference” hosts several papers on ITS for collaborative learning, covering adaptive feedback mechanisms, group modeling, and social learning. Overall, these studies have underlined the potential of ITS for collaborative learning, with an emphasis on adaptive feedback mechanisms, group modeling, and social learning. They also shed light on the hurdles and lessons drawn from the development and implementation of ITS in diverse educational environments. Our work extends these findings by exploring the influence of requested tutoring materials on student performance.

### 2.2 Skill Mastery and End-of-Unit Performance

Understanding the relationship between skill mastery during in-unit problems and subsequent performance in end-of-unit assignments has been a focal point of research in the field of educational data mining [30]. The advent of data-intensive learning environments, such as intelligent tutoring systems and adaptive learning platforms, has fueled this research. The granularity of data available from these platforms has allowed researchers to meticulously analyze student behaviors and learning trajectories [31]. Primarily, studies have demonstrated a strong correlation between students' mastery of specific skills during in-unit problems and their performance on end-of-unit assignments [32]. The underlying principle is the 'mastery learning' theory, which asserts that when students deeply understand concepts in a given unit, they can effectively apply this knowledge in subsequent, often more complex, tasks [33].

However, the process of skill mastery in learning is dynamic and iterative [34]. Learners often exhibit a 'staircase' behavior, characterized by phases of quick learning followed by periods of slower progress, which reflects the concept of 'zone of proximal development' [35]. As such, an accurate prediction of end-of-unit performance must consider not just the level of skill mastery but also the trajectory and pattern of learning. Further, the nature of the skills being learned is also crucial. Research suggests that 'hard skills'—those involving clear rules and procedures, such as mathematical operations, may demonstrate a different relationship between in-unit mastery and end-of-unit performance compared to 'soft skills'—- those involving interpretation and judgment, such as reading comprehension [36]. Therefore, while the correlation between in-unit skill mastery and end-of-unit performance is well-established, it is critical to consider the dynamism of the learning process, the type of skills involved, and the unique learning patterns of individual students in our exploration of this relationship.

Our study extends this line of research by investigating this relationship in the context of the ASSISTments platform and the Common Core State Standards (CCSS) aligned curriculum [37]. The ASSISTments platform, with its rich data on student interactions and performance, provides a fertile ground for such investigations [38]. The CCSS-aligned curriculum, with its focus on skill mastery and learning progressions, further supports the exploration of this relationship [39].

### 2.3 Feature Extraction and Predictive Modeling

Identifying relevant features from a rich dataset and understanding their importance in predicting student performance is another significant area of research in educational data mining [31]. This field has significantly benefited from the advent of machine learning, which provides powerful algorithms to recognize complex patterns and predict outcomes based on a large number of features. In the context of online learning environments, a wide variety of data can be collected, from simple metrics such as time spent on tasks and number of attempts [40] to more sophisticated measures like clickstream data [41] and textual data from student interactions [42]. Such rich datasets have the potential to provide a holistic view of the student's learning process, capturing both cognitive and affective aspects of learning.

Researchers have applied machine learning techniques to predict student performance using features extracted from these data types. For instance, classification methods like decision trees, random forests, and support vector machines have been employed to categorize students into different performance groups based on their learning behaviors [43]. Regression models have been used to predict quantitative measures of performance, such as final course grades [14]. Meanwhile, deep learning techniques are being explored for their ability to learn high-level representations from raw data, providing insights that were not possible with traditional methods [44].

Yet, the feature extraction process is not straightforward. It requires domain knowledge to identify relevant features and to interpret their importance in the predictive model. A challenge in this process is dealing with the high dimensionality of data, which could lead to overfitting. This issue is typically addressed through feature selection or dimensionality reduction techniques [45]. Hence, while predictive modeling has the potential to greatly enhance our understanding and prediction of student performance, it also calls for careful consideration of feature extraction and selection, model choice, and model interpretation.

Our study builds on these methods by leveraging an extensive set of features extracted from various dataset attributes in predicting end-of-unit assignment grades on the ASSISTments platform.

## 2.4 Graph Representation Learning in Online Learning

Graph representation learning has increasingly been applied to educational data to understand student behaviors and predict their performance [18]. [46] presented a flexible graph-structured model for predicting students' academic performance. The proposed model is a graph convolutional network (GCN) that considers the complex structures of undergraduate degree programs. [47] proposed a novel approach called R2GCN, which uses GNNs to model the relationship between students and questions using student interactions to construct the student-interaction-question network for generalizable student performance prediction in interactive online question pools. Study-GNN [16] is a pipeline for student performance prediction. The authors constructed multiple graphs based on different similarity measures between students' characteristics. Then they applied a multi-topology graph neural network (MTGNN) to classify students' performance into pass/fail or pass/withdrawal categories. [15] proposed DOPE, which models students' interactions with different courses as a knowledge graph and then uses relational graph neural networks (RGCNs) to learn latent representations for students. Simultaneously, they used an LSTM that encodes the temporal student behaviors. The two representations were combined to predict the student's performance at different points during the semester. [48] construct a graph based on the similarity between students' data and use a GCN as well as node2vec to create a low-dimensional representation for students. They enriched their original dataset with the learned representations to identify "at-risk" students. Our study contributes to this line of research by applying these techniques to the rich dataset provided by the ASSISTments platform, providing a nuanced understanding of student behaviors and improving the accuracy of grade predictions.

In conclusion, while our work builds on a rich body of research in educational data mining and learning analytics, it also makes several unique contributions. By examining the role of tutoring materials, investigating the relationship between skill mastery and assignment performance, conducting extensive feature extraction and analysis, and applying graph representation learning techniques, our study contributes to the theoretical understanding and practical application of data mining techniques in online learning contexts.

# CHAPTER 3

# DATASET

In this chapter, we first provide an overview of the dataset from the EDM Cup 2023 Kaggle Competition [49], detailing the tables and their salient attributes. Subsequently, we conduct on an initial exploration of the data, wherein we present various statistical insights derived from the dataset.

## 3.1 An Overview of the Dataset

The database schema for the dataset is demonstrated in Figure 3.1. Next, we describe each table and its notable fields shown in Figure 3.1.

- **assignment_details:** Each row in this table represents an assignment, including the unit test assignments, initiated by a student. These rows record the assignment of specific problem sequences to individual students.

- **sequence_details:** Each sequence present in the dataset is represented by at least one row in this table. Each row encapsulates a problem set comprising a sequence. The sequences, which are organized into folders in the original dataset, have their folder path levels mirrored in this table. The folder paths denote various attributes such as curriculum, grade, or subject associated with the sequence.

- **problem_details:** This table includes one row for each problem in the dataset, excluding some problems that have been deleted. One notable field is the Common Core State Standards (CCSS) for Mathematics [9] skill code, which is pertinent to the solution of the corresponding problem. The table also includes the first 32 principal components of the BERT embedding for the problem's text-based content, accurate to the 8th decimal place.

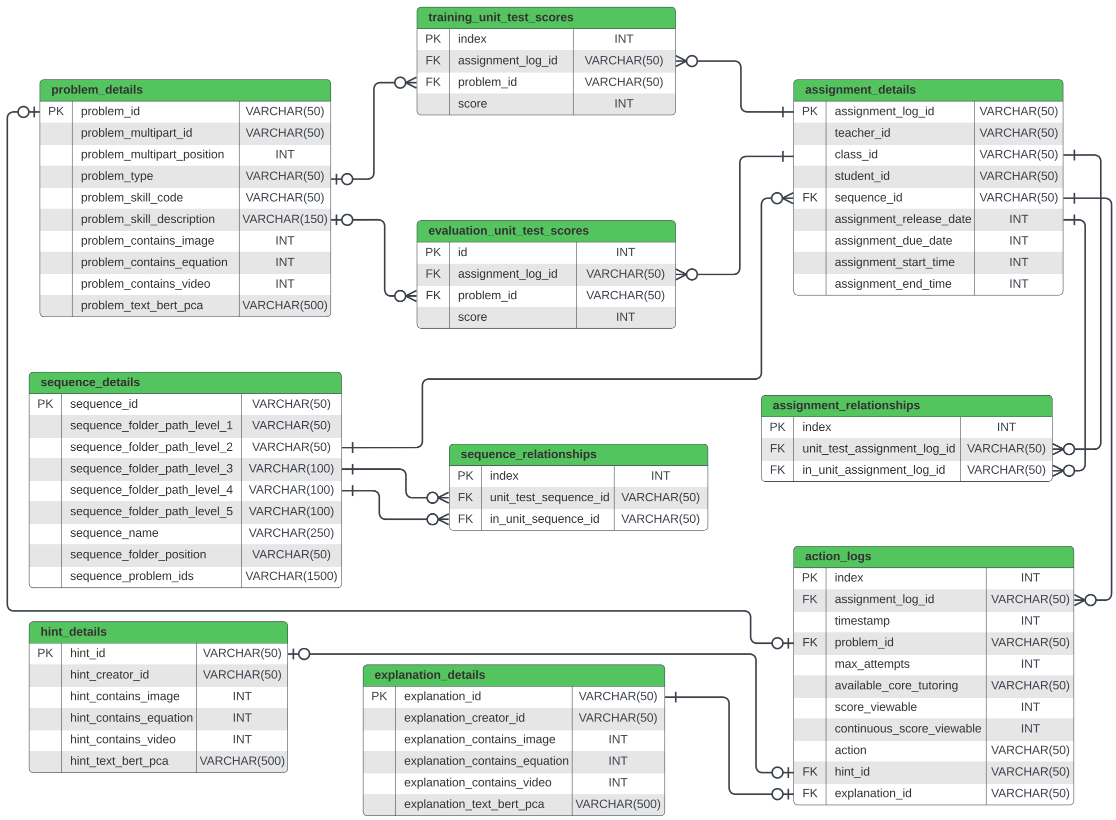


Fig. 3.1: Database schema of the data from the EDM Cup 2023 Kaggle Competition.

- **action_logs:** This table holds clickstream data, capturing student interactions with assignments. Specific problem interactions have corresponding unique identifiers, and initial interactions record data about tutoring availability, type, and maximum problem-solving attempts. The action field logs student interactions, indicating the correctness of responses and any request for hints or explanations. In the case of hint/explanation requests, their unique identifiers are recorded as well.

- **hint_details:** Except for some deleted hints, each hint in the dataset has a corresponding row in this table. It includes the first 32 principal components of the BERT embedding for the hint's text-based content, accurate to the 8th decimal place.

- **explanation_details:** Each explanation in the dataset is represented by a row in this table. It includes the first 32 principal components of the BERT embedding for the explanation's text-based content, accurate to the 8th decimal place.

- **training_unit_test_scores:** This table contains unit test scores used for training the score prediction model. The scores are binary; 1 denotes completion of an open-ended response problem or correct first-attempt response without tutoring, while 0 signifies otherwise.

- **evaluation_unit_test_scores:** This table includes the unit test assignment log IDs, which will be used for score prediction in model evaluation.

- **assignment_relationships:** This table specifies which assignments within the unit correspond to the unit test assignments in the training and evaluation sets.

- **sequence_relationships:** This table details the sequences that are unit tests and the corresponding sequences within the unit.

### 3.2 Initial Data Exploration

Exploring the initial data is the first step towards analyzing and modeling the dataset, enabling us to understand its characteristics and structure. Table 3.1 presents the basic statistics for all entities in the dataset. Additionally, Figure 3.2 illustrates the distribution of scores for all end-of-unit assignments, indicating that the dataset is slightly imbalanced in favor of problems with score 1. Furthermore, Figure 3.3 presents the average score per grade. Grades are determined based on the first part of the problem skill code, which will be further explained in detail in Chapter 4.2. **HSS** stands for High School Statistics and Probability, **HSF** for High School Functions, **HSG** for High School Geometry, **HSN** for High School Number and Quantity, and **HSA** for High School Algebra. The plot indicates that grade 1 students achieved the highest average score, whereas grade 7 students obtained the lowest average score. Moreover, Figure 3.4 illustrates the number of students who participated in the exam. Due to the absence of information connecting students to their respective grades in the dataset, it was not possible to categorize all 651,253 students based on their grades.

Table 3.1: Basic statistics of the dataset

| Entity | Count |
|---|---|
| # Students | 651,253 |
| # Teachers | 23,523 |
| # Classes | 47,401 |
| # Sequences | 10,228 |
| # Problems | 132,738 |
| # Assignments | 9,319,676 |
| # Hints | 8,381 |
| # Explanations | 4,132 |
| # Problems per sequence | $\mu$=13 |
| # Unfinished assignments | 1,878,016 |

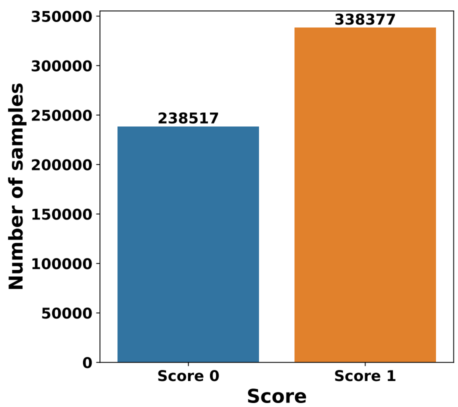


Fig. 3.2: Score distribution

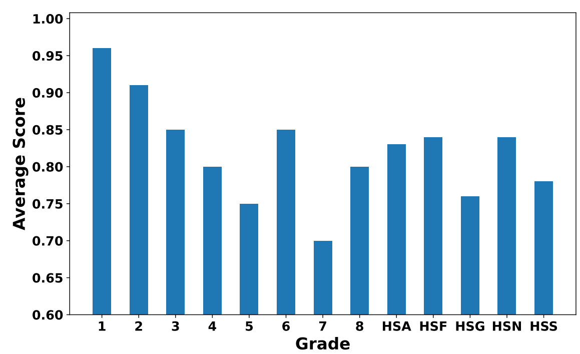


Fig. 3.3: Average score per grade

Figure 3.5 shows the number of problems for each problem type including both in-unit and end-of-unit problems. The most common problem type in the dataset is 'Number', followed by 'ungraded open response'. However, it is important to note that the latter type is only available during in-unit assignments and is not included in the end-of-unit assignments. Next, we investigated the core tutoring options available for in-unit problems using the action logs data. Figure 3.6 depicts the distribution of these tutoring options, revealing that approximately 40% of problems have no available tutoring. In Chapter 4.1, we will delve deeper into whether these in-unit tutoring options enhance students' performance in end-of-unit assignments. Additionally, we explored the sequence details table (explained in Chapter 3.1) to obtain average scores by grouping grades (sequence folder path level 2) and topics (sequence folder path level 3). This analysis allowed us to identify the top

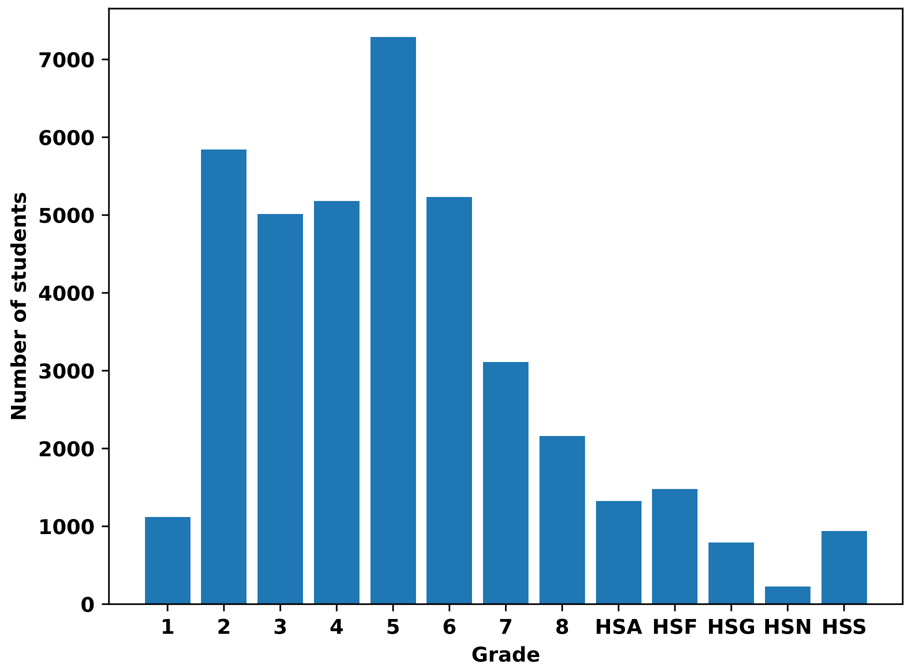


Fig. 3.4: Distribution of exam takers across grades

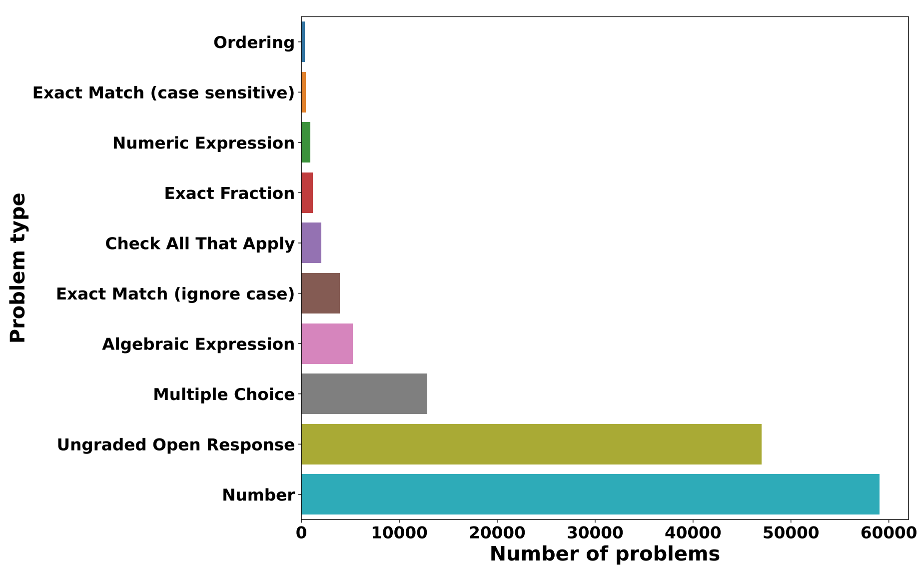


Fig. 3.5: Number of problems of each problem type

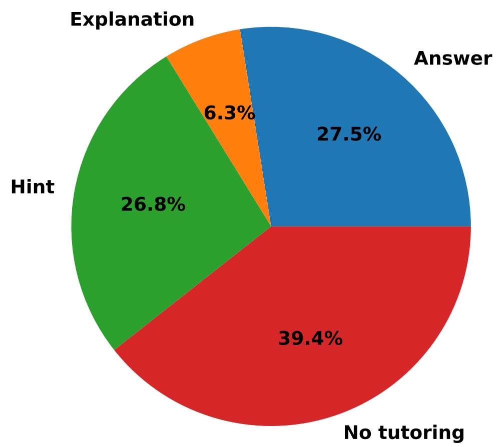


Fig. 3.6: Distribution of core tutoring options available for problems within the units.

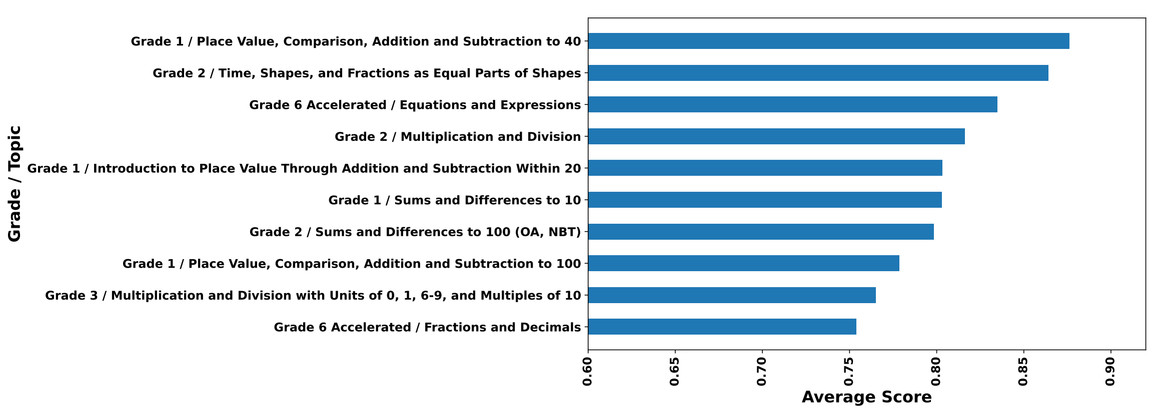


Fig. 3.7: Top 10 topics based on average scores of end-of-unit assignments

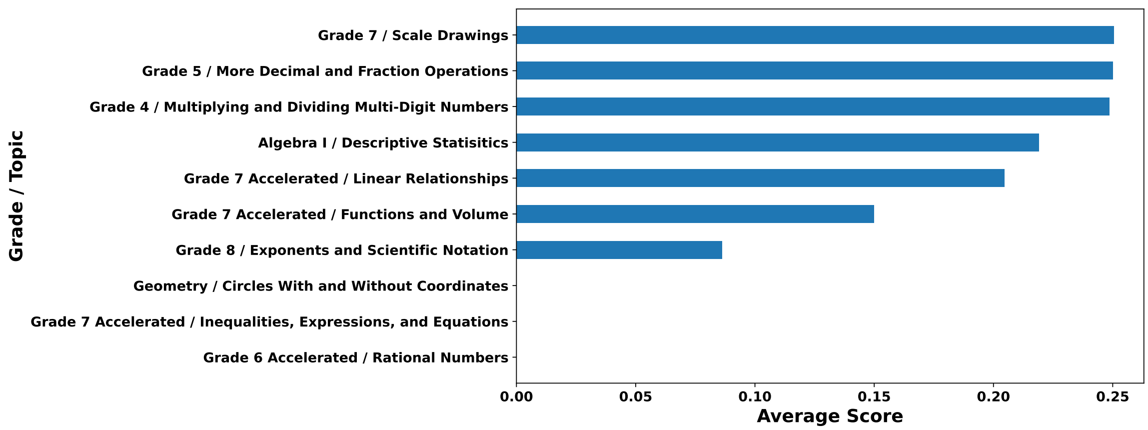


Fig. 3.8: Bottom 10 topics based on average scores of end-of-unit assignments

10 and bottom 10 topics in terms of average scores, revealing the topics where students are struggling and excelling. Figures 3.7 and 3.8 present the top and bottom 10 topics, respectively, based on their average scores. Figure 3.8 clearly indicates that students are facing difficulties with topics such as 'Rational Numbers', 'Inequalities, Expression and Equations', and 'Circles With and Without Coordinates', as these topics have an average score of zero. In Chapter 4.1, we dig deeper into patterns revealing the relationship between the different topics triangulated with student performance.

# CHAPTER 4

# STUDENT BEHAVIOR ANALYSIS AND ACADEMIC ACHIEVEMENT

To address **RQ1** and **RQ2**, we embark on a comprehensive analysis of student behaviors and their impact on academic outcomes. In Chapter 4.1, we explore the relationship between tutoring requests and the end-of-unit student performance, considering various tutoring alternatives. Following this, in Chapter 4.2, we employ association rule mining to extract meaningful patterns (rules). These rules serve to illuminate the relationship between two key concepts: a) the mastery or non-mastery of CCSS-related skills during in-unit assignments and b) the mastery or non-mastery of CCSS-related skills during end-of-unit problems.

## 4.1 Tutoring Request and Student Performance

When students engage in in-unit assignments, they have the option to request tutoring if it is available for the specific problem they are working on. The available core tutoring options include *hints*, *explanations*, and *answers*. Additionally, there are two auxiliary tutoring options including *skill-related videos*, and *live tutor*. We aim to assess the effectiveness of these tutoring options in improving student performance in end-of-unit problems. To this end, we calculated the percentage of each tutoring option requested for each group of end-of-unit assignment log ID and problem ID, which are associated with multiple action log problems. The percentage of tutoring requested is determined by dividing the number of problems for which students requested a particular tutoring option by the total number of problems where that specific tutoring option was available. This is expressed by the following formula:

$$\text{\% of tutoring option requested} = \frac{\text{\# of problems with tutoring option requested}}{\text{Total \# of problems with that tutoring option available}} \times 100 \tag{4.1}$$

Table 4.1: % Hint Requested (HR)

| **Grade** | **HR** ($\geq 0.7$) | **HR** ($< 0.3$) | **SD** | **t-stat** | **p-value** |
|---|---|---|---|---|---|
| 1 | 0.65 | 0.86 | -0.21 | -2.89 | $3.93e{-3}$ |
| 2 | 0.41 | 0.71 | -0.30 | -4.63 | $3.65e{-6}$ |
| 3 | 0.21 | 0.60 | -0.39 | -11.00 | 0.00 |
| 4 | 0.42 | 0.53 | -0.11 | -6.52 | $6.93e{-11}$ |
| 5 | 0.40 | 0.49 | -0.09 | -2.99 | $2.79e{-3}$ |
| 6 | 0.34 | 0.59 | -0.25 | -30.53 | 0.00 |
| 7 | 0.38 | 0.56 | -0.18 | -25.03 | 0.00 |
| 8 | 0.45 | 0.54 | -0.09 | -7.10 | $1.27e{-12}$ |
| HSA | 0.48 | 0.71 | -0.23 | -2.78 | $5.52e{-3}$ |
| HSF | 0.00 | 0.63 | -0.63 | -1.84 | $6.72e{-2}$ |
| HSG | 1.00 | 0.79 | 0.21 | 2.01 | $4.56e{-2}$ |
| HSN | 0.33 | 0.61 | -0.28 | -2.09 | $3.88e{-2}$ |
| HSS | 0.25 | 0.60 | -0.35 | -1.41 | $1.61e{-1}$ |
| **All grades** | 0.38 | 0.61 | -0.22 | -51.93 | 0.00 |

Table 4.2: % Answer Requested (AR)

| **Grade** | **AR** ($\geq 0.7$) | **AR** ($< 0.3$) | **SD** | **t-stat** | **p-value** |
|---|---|---|---|---|---|
| 1 | 0.76 | 0.78 | -0.01 | -0.84 | $3.99e{-1}$ |
| 2 | 0.60 | 0.77 | -0.17 | -50.74 | 0.00 |
| 3 | 0.45 | 0.67 | -0.22 | -45.56 | 0.00 |
| 4 | 0.40 | 0.67 | -0.27 | -56.94 | 0.00 |
| 5 | 0.32 | 0.61 | -0.29 | -69.48 | 0.00 |
| 6 | 0.48 | 0.67 | -0.19 | -43.93 | 0.00 |
| 7 | 0.45 | 0.66 | -0.21 | -31.66 | 0.00 |
| 8 | 0.41 | 0.61 | -0.20 | -26.69 | 0.00 |
| HSA | 0.36 | 0.69 | -0.33 | -20.55 | 0.00 |
| HSF | 0.37 | 0.69 | -0.32 | -23.96 | 0.00 |
| HSG | 0.34 | 0.61 | -0.27 | -18.82 | 0.00 |
| HSN | 0.40 | 0.64 | -0.24 | -5.39 | $9.56e{-8}$ |
| HSS | 0.35 | 0.49 | -0.14 | -5.71 | $1.25e{-8}$ |
| **All grades** | 0.47 | 0.69 | -0.22 | -145.87 | 0.00 |

Table 4.3: % Explanation Requested (ER)

| **Grade** | **ER** ($\geq 0.7$) | **ER** ($< 0.3$) | **SD** | **t-stat** | **p-value** |
|---|---|---|---|---|---|
| 5 | 0.17 | 0.41 | -0.24 | -2.35 | $1.91e{-2}$ |
| 6 | 0.27 | 0.64 | -0.37 | -23.92 | 0.0 |
| 7 | 0.33 | 0.58 | -0.25 | -15.36 | 0.0 |
| 8 | 0.32 | 0.55 | -0.23 | -9.25 | 0.0 |
| **All grades** | 0.30 | 0.61 | -0.31 | -29.84 | $9.47e{-194}$ |

Table 4.4: % Skill Video Requested (SVR)

| **Grade** | **SVR** ($\geq 0.7$) | **SVR** ($< 0.3$) | **SD** | **t-stat** | **p-value** |
|---|---|---|---|---|---|
| 2 | 0.83 | 0.77 | 0.07 | 0.68 | 0.50 |
| 4 | 0.64 | 0.58 | 0.05 | 0.63 | 0.53 |
| 5 | 0.43 | 0.50 | -0.07 | -1.31 | 0.19 |
| 6 | 0.63 | 0.54 | 0.09 | 1.11 | 0.27 |
| 7 | 0.12 | 0.53 | -0.41 | -4.22 | 0.00 |
| 8 | 0.50 | 0.54 | -0.04 | -0.35 | 0.73 |
| HSF | 0.75 | 0.64 | 0.11 | 0.62 | 0.54 |
| **All grades** | 0.52 | 0.54 | -0.03 | -0.79 | 0.43 |

Table 4.5: % Live Tutor Requested (LTR)

| **Grade** | **LTR** ($\geq 0.7$) | **LTR** ($< 0.3$) | **SD** | **t-stat** | **p-value** |
|---|---|---|---|---|---|
| 6 | 0.00 | 0.54 | -0.54 | - | - |
| 7 | 0.86 | 0.53 | 0.33 | 4.01 | $6.13e{-5}$ |
| 8 | 0.80 | 0.54 | 0.26 | 1.16 | $2.47e{-1}$ |
| HSF | 0.89 | 0.65 | 0.24 | 4.67 | $4.36e{-6}$ |
| **All grades** | 0.87 | 0.54 | 0.33 | 7.83 | $5.03e{-15}$ |

Table 4.6: % Total Tutoring Requested (TTR)

| **Grade** | **TTR** ($\geq 0.7$) | **TTR** ($< 0.3$) | **SD** | **t-stat** | **p-value** |
|---|---|---|---|---|---|
| 1 | 0.29 | 0.78 | -0.50 | -4.53 | $6.02e{-6}$ |
| 2 | 0.34 | 0.72 | -0.38 | -26.24 | 0.0 |
| 3 | 0.25 | 0.62 | -0.37 | -20.15 | 0.0 |
| 4 | 0.27 | 0.58 | -0.31 | -28.58 | 0.0 |
| 5 | 0.32 | 0.55 | -0.23 | -20.47 | 0.0 |
| 6 | 0.27 | 0.61 | -0.34 | -32.19 | 0.0 |
| 7 | 0.31 | 0.58 | -0.27 | -26.80 | 0.0 |
| 8 | 0.30 | 0.54 | -0.25 | -13.78 | 0.0 |
| HSA | 0.24 | 0.60 | -0.36 | -4.30 | $1.72e{-5}$ |
| HSF | 0.00 | 0.60 | -0.60 | -1.75 | $8.09e{-2}$ |
| HSG | 0.44 | 0.54 | -0.10 | -2.66 | $7.79e{-3}$ |
| HSN | 0.25 | 0.59 | -0.34 | -3.05 | $2.35e{-3}$ |
| HSS | 0.43 | 0.45 | -0.02 | -0.12 | $9.06e{-1}$ |
| **All grades** | 0.31 | 0.63 | -0.32 | -73.61 | 0.00 |

Tables 4.1 to 4.5 present the results of our examination of the association between students' end-of-unit performance and their requests for the five distinct tutoring options available. Table 4.6 consolidates the results for all tutoring options. Additionally, we conducted a grade-specific analysis by segmenting the data based on the initial segment of the problem skill code, which corresponds to the grade as per the CCSS guidelines. This approach facilitated the computation of the proportion of tutoring requests for each grade and tutoring option independently. However, not all tutoring options were accessible for every grade or had sufficient data to enable experimentation, leading to variations across tables. The second and third columns display the average grade of problems where the proportion of the tutoring option, computed using Eq. 4.1, exceeds 70% and falls below 30%, respectively. The selected percentages aim to yield a stark contrast, allowing for a more confident assessment of the impact of tutoring requests. It is worth noting that despite the appeal of considering tutoring requests in a binary fashion (i.e., 100% and 0%), the scarcity of cases where a student consistently requests or refrains from requesting tutoring makes it an impractical approach for addressing **RQ1**. The fourth column, denoted as **SD**, represents the score difference between the second and third columns. For example, for grade 1 with hint requests, **SD** = -0.21 = **HR** ($\geq$ 0.7) - **HR** ($<$ 0.3). This difference helps to evaluate whether requesting a tutoring option frequently within the unit yields a better grade on the unit test (**SD** $\geq$ 0) or not (**SD** $<$ 0). The fifth column showcases the t-statistic, derived via a Student's t-test. As a measure used in hypothesis testing, the t-statistic determines the likelihood of the observed difference between sample means having occurred by chance, thereby assisting in establishing statistical significance [50]. The final columns exhibit the p-value. The outcomes of these analyses are illustrated in Tables 4.2 to 4.6. We make the following observations based on these results.

- In general, students who requested help less often (HR, AR, ER, SVR, LTR $<$ 0.3) performed better than those who requested more often ($\geq$ 0.7). This is observed in all grades for HR, AR, ER, and TTR, in grades 2, 5, and 7 for SVR, and in grade 7 and HSF for LTR.

- ➪ The largest difference in performance (SD) between the two groups is observed in Explanation Requested (ER) for grade 6, Skill Video Requested (SVR) for grade 7, and Total Tutoring Requested (TTR) for grade 1.

- ➪ The t-statistic is significant (p-value $< 0.05$) for most cases across all tutoring options, indicating a significant difference in the means of the two groups (those who request help more often vs. those who request less often).

- ➪ In the case of Hint Requested (HR), there is a reverse trend in HSG, where those who requested more often performed better. A similar reverse trend is observed in grade 7 and HSF for Live Tutor Requested (LTR).

- ➪ In the Skill Video Requested (SVR) section, there is no significant difference (p-value $> 0.05$) between the performance of those who requested help more often vs. less often in grades 2, 4, 6, 8, and HSF.

- ➪ In the case of Live Tutor Requested (LTR), there is a large difference (SD $= 0.54$) in grade 6, but the t-statistic is not reported, suggesting an insufficient sample size or other statistical issue.

- ➪ Across all grades, students who requested help less often performed better in all tutoring options.

Additionally, we explored the percentage of correct and wrong responses in the action logs of in-unit assignments to assess their impact on student's grades in end-of-unit problems. Table 4.7, 4.8 show the results. The data from Tables 4.7, 4.8 offer several key observations about the impact of students' correct and wrong response rates on in-unit assignments on their final grades.

- ➪ For the percentage of correct responses (CR), students with a CR of 0.7 or higher consistently have a higher grade than those with a CR of less than 0.3. This trend is universal across all grades, with the highest difference seen in grade 2, where students with higher CR had a 0.25 higher grade. It's also important to note that the p-values

Table 4.7: % Correct Response (CR)

| **Grade** | **CR** ($\geq 0.7$) | **CR** ($< 0.3$) | **SD** | **t-stat** | **p-value** |
|---|---|---|---|---|---|
| 1 | 0.79 | 0.64 | 0.15 | 8.83 | 0.0 |
| 2 | 0.75 | 0.50 | 0.25 | 31.28 | 0.0 |
| 3 | 0.61 | 0.48 | 0.12 | 9.13 | $7.30e{-20}$ |
| 4 | 0.54 | 0.35 | 0.19 | 17.19 | 0.0 |
| 5 | 0.51 | 0.42 | 0.09 | 12.78 | 0.0 |
| 6 | 0.51 | 0.48 | 0.04 | 4.14 | $3.56e{-5}$ |
| 7 | 0.56 | 0.30 | 0.26 | 19.55 | 0.0 |
| 8 | 0.50 | 0.40 | 0.09 | 7.89 | $3.48e{-15}$ |
| HSA | 0.61 | 0.43 | 0.18 | 7.26 | $6.06e{-13}$ |
| HSF | 0.58 | 0.52 | 0.06 | 2.57 | $1.03e{-2}$ |
| HSG | 0.52 | 0.50 | 0.03 | 1.40 | $1.62e{-1}$ |
| HSN | 0.71 | 0.43 | 0.28 | 3.68 | $2.72e{-4}$ |
| HSS | 0.55 | 0.43 | 0.13 | 3.43 | $6.33e{-4}$ |
| **All grades** | 0.62 | 0.45 | 0.17 | 54.11 | 0.00 |

Table 4.8: % Wrong Response (WR)

| **Grade** | **WR** ($\geq 0.7$) | **WR** ($< 0.3$) | **SD** | **t-stat** | **p-value** |
|---|---|---|---|---|---|
| 1 | 0.57 | 0.81 | -0.24 | -11.47 | 0.0 |
| 2 | 0.50 | 0.75 | -0.25 | -39.49 | 0.00 |
| 3 | 0.39 | 0.64 | -0.25 | -35.47 | 0.0 |
| 4 | 0.39 | 0.62 | -0.23 | -36.92 | 0.0 |
| 5 | 0.38 | 0.57 | -0.19 | -38.84 | 0.00 |
| 6 | 0.41 | 0.63 | -0.23 | -40.95 | 0.00 |
| 7 | 0.38 | 0.60 | -0.22 | -29.70 | 0.0 |
| 8 | 0.41 | 0.55 | -0.14 | -14.31 | 0.0 |
| HSA | 0.40 | 0.61 | -0.21 | -8.63 | $8.41e{-18}$ |
| HSF | 0.47 | 0.63 | -0.17 | -6.71 | $2.14e{-11}$ |
| HSG | 0.35 | 0.61 | -0.26 | -18.11 | 0.0 |
| HSN | 0.50 | 0.63 | -0.13 | -2.24 | $2.55e{-2}$ |
| HSS | 0.29 | 0.49 | -0.20 | -6.31 | $3.25e{-10}$ |
| **All grades** | 0.42 | 0.67 | -0.24 | -114.70 | 0.00 |

for these differences are all significantly less than 0.05, suggesting these results are statistically significant.

- On the other hand, for the percentage of wrong responses (WR), students with a WR of less than 0.3 consistently have higher grades than those with a WR of 0.7 or higher. The difference is most pronounced in grade 3, where students with lower WR had a 0.25 higher grade. As with the CR, the p-values for these differences are all significantly less than 0.05, confirming these results are statistically significant.

- In summary, students with higher CR percentages tend to perform better, while those with higher WR percentages are more likely to struggle. This suggests that the frequency of correct and wrong responses during in-unit assignments could be a strong predictor of a student's final grade.

The analysis of this part of the thesis indicates that students who request tutoring less frequently, in most cases, perform better, potentially suggesting a higher degree of self-reliance and capability to resolve problems independently. However, a notable exception is the positive impact of live tutor requests (LTR) on performance, implying that direct tutor interactions can be advantageous. Despite these correlations, causation should not be prematurely assumed as the observations are context-specific. On the other hand, if a student regularly requests assistance through hints, explanations, and answers or frequently gives incorrect responses, it suggests that they may be grappling with the subject matter. Such patterns could signal the need for additional teacher support [51],targeted interventions [52], or personalized assistance [53] to bolster their comprehension and academic performance. Therefore, discerning and addressing these struggle areas is crucial for educators to provide suitable guidance and resources, thereby enhancing the chances of student success.

### 4.2 CCSS Skill Mastery and Student Performance

Analyzing the relationship between mastery/non-mastery of skills during in-unit assignments and end-of-unit assignments is crucial. It enables educators to understand which skills mastered/non-mastered during in-unit problems contribute to success/failure in corresponding end-of-unit problems [38, 54, 55]. It is important to recognize that certain skills require prerequisite knowledge for comprehension. Therefore, if a student fails to grasp the prerequisite skills or lacks practice in in-unit problems related to those skills, it may result in failure to solve the corresponding skill problems in end-of-unit assignments. On the other hand, success in mastering prerequisite skills through in-unit problems will likely lead to success in related end-of-unit problems. Thus, it is crucial to analyze these patterns in student performance data and identify relationships between different mathematical skills. Success or failure in in-unit problems requires having scores for these problems. However, unlike end-of-unit problems, since the dataset does not provide scores for in-unit problems, we needed to determine these scores. Next, we describe how we addressed this challenge.

We combined the training and action log tables to gather all in-unit problems associated with end-of-unit problems– See Chapter 5.1. Then, we specified the scores for in-unit problems based on the scoring criteria defined for end-of-unit problems as explained in Chapter 3. More specifically, this process involves considering the 'action' feature while disregarding common actions such as 'assignment_started,' 'problem_started,' 'problem_finished,' 'continue_selected,' 'assignment_finished,' and 'assignment_resumed.' After excluding these common actions, eight actions remained: 'wrong_response,' 'answer_requested,' 'correct_response,' 'open_response,' 'skill_related_video_requested,' 'explanation_requested,' 'hint_requested,' and 'live_tutor_requested'. Finally, the score of an in-unit problem is determined according to Algorithm 1:

Once we determined the grades for in-unit problems, we needed to extract and fix skill codes since. Note however, we are interested in mastery/non-mastery for *skills*, not specific problems. The CCSS skill codes follow a hierarchical structure, where the first level corresponds to the grade, and the second level represents the topic or subject. The later levels of a CCSS skill code provide more specific descriptions, such as specific problem details. However, these detailed levels may not be as useful since they can potentially generate skill-related patterns that are overly specific. Therefore, for the purpose of generating

**Algorithm 1** In-unit score specification

**Require:** in_unit_assignment_log_id: log ids of in-unit assignments
**Require:** problem_id: problem ids of in-unit assignments
**Require:** action: action feature of in-unit assignment logs
**Ensure:** score: The score assigned to the problems based on action feature.

```
for all in_unit_assignment_log_id, problem_id do
  for action in actions do
    if action = 'open_response' then
      score = 1
    else if action in ['Wrong_response', 'hint_requested', 'explanation_requested',
    'live_tutor_requested', 'skill_related_video_requested', 'answer_requested'] then
      if score = Null then
        score = 0
      end if
    else if action 'correct_response' then
      if score = Null then
        score = 1
      end if
    end if
  end for
end for
```

meaningful patterns, we focused on the first two levels of the skill code hierarchy, which capture the broader grade and topic information, respectively.

So far, we have grades (success or failure) for all in-unit and end-of-unit problems as well as their corresponding skill codes. Now, we need to specify how we can extract meaningful "patterns". One effective approach for achieving this is by utilizing association rule learning/mining, which is a data mining technique employed to discover interesting relationships or patterns within extensive datasets [10, 11] such as educational data [56]. They identify frequently occurring *itemsets* and generate rules that describe associations between different items based on their co-occurrence. By leveraging these rules, educators can gain valuable insights into patterns and dependencies among skills. To fix the idea, in the following, we formally define the association rule mining process on the CCSS skill codes.

Let $\mathcal{C} = \{C_1, C_2 \cdots C_k\}$ denote a set of CCSS skill code levels e.g., *the Complex Number System*. Suppose, $\mathcal{P}^s = \{p^s_1, p^s_2 \cdots p^s_{n_1}\}$ is the set of end-of-unit problems ($|\mathcal{P}^s| = n_1$), for which the score is 1 i.e., *success*. Correspondingly, $\mathcal{P}^f = \{p^f_1, p^f_2 \cdots p^f_{n_2}\}$ is the set of end-of-unit problems ($|\mathcal{P}^f| = n_2$), for which the score is 0 i.e., *failure*. For each $p^s_i \in P^s$, we define a *transaction* as $\mathbf{I}^i = \{I_1, I_2 \cdots I_{m_i-1}, I_{m_i}\}$ where each $I_j \in \mathcal{C}$, $I_{m_i}$ is the skill code of the end-of-unit problem $p^s_i$, and $I_1, I_2 \cdots I_{m_i-1}$ are skill codes of the *related* in-unit problems. In other words, each transaction includes the skill codes of both the end-of-unit problem and all related in-unit problems. The score of all problems in $P^s$ is 1 (success), where the scores of in-unit problems are calculated using Algorithm 1. Let $\mathcal{I}^s = \{\mathbf{I}^1, \mathbf{I}^2 \cdots \mathbf{I}^{n_1}\}$ represent all $n_1$ transactions pertinent to $\mathcal{P}^s$. In a similar manner, we define transaction set $\mathcal{I}^f$ pertinent to $\mathcal{P}^f$. Furthermore, we divided the datasets for each grade (first level of skill code) to ensure that item sets or transactions are categorized by grade. Based on the above notations, we define association rule mining for skill mastery and non-mastery rule discovery.

**Skill Mastery Rule Discovery** Given $\mathcal{I}^s$, we are interested in strong rules in the form of $X \rightarrow y$, where $X \subset \mathcal{C}$ and $y \in \mathcal{C}$. $y$ is a single item representing the skill code of the end-of-unit problem.

**Skill Non-Mastery Rule Discovery** Given $\mathcal{I}^f$, we are interested in strong rules in the form of $X \rightarrow y$, where $X \subset \mathcal{C}$ and $y \in \mathcal{C}$. $y$ is a single item representing the skill code of the end-of-unit problem.

In association rule learning, "strong" rules are captured by two rule-related concepts: support and confidence. **Support** (**S**) is the proportion of transactions in the dataset that contain a particular itemset. For a rule $X \rightarrow Y$, it can be defined as:

$$\text{Support}(X \cup Y) = \frac{\text{Frequency}(X \cup Y)}{\text{Total number of transactions}} \tag{4.2}$$

**Confidence (C)** is a measure of the reliability of the inference made by a rule. For a rule $X \rightarrow Y$, it can be defined as:

$$\text{Confidence}(X \rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)} = \frac{\text{Number of transactions containing X and Y}}{\text{Number of transactions containing X}} \tag{4.3}$$

To extract rules, we used *mlxtend* python library [57], which uses the famous Apriori algorithm [58,59] for efficient rule mining. To generate frequent item sets, we identify items that occur frequently by setting a minimum support threshold of 0.8 for mastery (score 1) and 0.7 for non-mastery (score 0). Subsequently, we derive association rules using the confidence metric, with a minimum threshold of 0.9 for mastery (score 1) and 0.8 for non-mastery (score 0). Tables 4.9 and 4.10 demonstrate the results of association rule learning for skill mastery and non-mastery, respectively. The first column is the rule, the second column is the support, the third column is the confidence, and the last column is the rule where the codes have been replaced with their English description. We obtained the descriptions for skill codes up to level 2 from CCSS [9] since they were not included in the dataset.

Tables 4.9 and 4.10 display the most reliable rules among all the transactions with high support and confidence. Based on the results in Tables 4.9 and 4.10, we make the following observations:

- ➪ The high support and confidence of most rules in both tables indicate that there are strong associations among the different math skills in the dataset. For example, the rule "8.EE → HSA.REI" in Table 4.9, with a support of 0.81 and confidence of 1.00, suggests that students who have mastered "Expressions and Equations" (8.EE) are also very likely to have mastered "Reasoning with Equations and Inequalities" (HSA.REI).

- ➪ The rule "HSN.RN, HSN.CN → HSA.REI" in Table 4.9 shows that the mastery of the Real Number System (HSN.RN) and the Complex Number System (HSN.CN) is highly associated with the mastery of Reasoning with Equations and Inequalities

Table 4.9: Extracted association rules for mastery (score 1) for the entire dataset

| Rule | S | C | Rule Description |
|---|---|---|---|
| 8.EE → HSA.REI | 0.81 | 1.00 | Expressions and Equations → Reasoning with Equations and Inequalities |
| HSN.RN → HSA.REI | 0.93 | 0.99 | The Real Number System → Reasoning with Equations and Inequalities |
| HSN.RN, HSN.CN → HSA.REI | 0.86 | 0.99 | The Real Number System, The Complex Number System → Reasoning with Equations and Inequalities |
| 8.EE → HSN.RN | 0.80 | 0.99 | Expressions and Equations → The Real Number System |
| HSN.CN → HSA.REI | 0.92 | 0.99 | The Complex Number System → Reasoning with Equations and Inequalities |
| HSA.REI → HSN.RN | 0.93 | 0.94 | Reasoning with Equations and Inequalities → The Real Number System |
| HSA.REI → HSN.CN | 0.92 | 0.93 | Reasoning with Equations and Inequalities → The Complex Number System |
| HSN.CN, HSA.REI → HSN.RN | 0.86 | 0.93 | The Complex Number System, Reasoning with Equations and Inequalities → The Real Number System |
| HSN.RN → HSN.CN | 0.87 | 0.93 | The Real Number System → The Complex Number System |
| HSN.CN → HSN.RN | 0.87 | 0.93 | The Complex Number System → The Real Number System |
| HSN.RN, HSA.REI → HSN.CN | 0.86 | 0.93 | The Real Number System, Reasoning with Equations and Inequalities → The Complex Number System |

Table 4.10: Extracted association rules for non-mastery (score 0) for the entire dataset

| Rule | S | C | Rule Description |
|---|---|---|---|
| HSA.REI → HSN.CN | 0.79 | 0.93 | Reasoning with Equations and Inequalities → The Complex Number System |
| HSN.RN, HSA.REI → HSN.CN | 0.71 | 0.93 | The Real Number System, Reasoning with Equations and Inequalities → The Complex Number System |
| HSN.RN → HSN.CN | 0.81 | 0.92 | The Real Number System → The Complex Number System |
| HSA.REI → HSN.RN | 0.77 | 0.91 | Reasoning with Equations and Inequalities → The Real Number System |
| HSN.CN, HSA.REI → HSN.RN | 0.71 | 0.90 | The Complex Number System, Reasoning with Equations and Inequalities → The Real Number System |
| HSF.BF → HSF.IF | 0.71 | 0.89 | Building Functions → Interpreting Functions |
| HSN.RN, HSN.CN → HSA.REI | 0.71 | 0.88 | The Real Number System, The Complex Number System → Reasoning with Equations and Inequalities |
| HSN.CN → HSN.RN | 0.81 | 0.88 | The Complex Number System → The Real Number System |
| HSN.RN → HSA.REI | 0.77 | 0.87 | The Real Number System → Reasoning with Equations and Inequalities |
| HSN.CN → HSA.REI | 0.79 | 0.85 | The Complex Number System → Reasoning with Equations and Inequalities |

(HSA.REI). This reflects the progressive complexity in learning mathematics, where advanced concepts often rely on the mastery of more basic concepts.

- There is one unique rule in the non-mastery table, “HSF.BF $\rightarrow$ HSF.IF”, which suggests that students who have not mastered “Building Functions” (HSF.BF) are also likely not to have mastered “Interpreting Functions” (HSF.IF). This could imply that the skills required for interpreting functions are contingent on the ability to build functions, or vice versa.

- Another observation is the high frequency of HSA.REI (Reasoning with Equations and Inequalities), HSN.RN (High School Number and Quantity - The Real Number System), and HSN.CN (The Complex Number System) in the rules, which implies that these concepts might be fundamental to mastery for High School subjects in this ASSISTments platform.

- Certain rules are evident in both Tables 4.9 and 4.10, including “The Real Number System $\rightarrow$ The Complex Number System”. Interpreting from Table 4.9, it is inferred that mastery in the real number system is a strong predictor for mastery in the complex number system. Conversely, Table 4.10 implies that students who struggle with the real number system are likely to face challenges in understanding the complex number system. These rules underscore the interdependency of understanding these two mathematical concepts.

- Finally, it is worth noting that while support and confidence are high for most rules, they are not absolute indicators of causality. The relationships could be affected by other factors such as the order of teaching, student demographics, etc. They do, however, offer a strong basis for further investigation into these associations.

The diversity of skill codes in non-high school grades means that the support of their corresponding rules is quite low and does not meet the minimum support threshold of 0.7. This explains why all the rules in Tables 4.9 and 4.10 pertain solely to high school grades. However, to empirically demonstrate the robustness of association rule mining for

Table 4.11: Top association rule for mastery (score 1) in each grade

| **Rule** | **S** | **C** | **Rule Description** |
|---|---|---|---|
| 1.NBT → 1.OA | 0.41 | 0.83 | Grade 1 - Number and Operations in Base Ten → Grade 1 - Operations and Algebraic Thinking |
| 2.OA → 2.NBT | 0.61 | 0.96 | Grade 2 - Operations and Algebraic Thinking → Grade 2 - Number and Operations in Base Ten |
| 3.NBT → 3.MD | 0.15 | 0.57 | Grade 3 - Number and Operations in Base Ten → Grade 3 - Measurement and Data |
| 4.OA → 4.NBT | 0.45 | 0.88 | Grade 4 - Operations and Algebraic Thinking → Grade 4 - Number and Operations in Base Ten |
| 5.MD → 5.NBT | 0.37 | 0.70 | Grade 5 - Measurement and Data → Grade 5 - Number and Operations in Base Ten |
| 4.NBT → 6.NS | 0.15 | 0.75 | Grade 4 - Number and Operations in Base Ten → Grade 6 - The Number System |
| 7.G → 7.RP | 0.47 | 0.81 | Grade 7 - Geometry → Grade 7 - Ratios and Proportional Relationships |
| 7.G → 8.G | 0.32 | 0.98 | Grade 7 - Geometry → Grade 8 - Geometry |
| HSA.CED → HSA.REI | 0.61 | 0.92 | High School Algebra - Creating Equations → High School Algebra - Reasoning with Equations and Inequalities |
| HSF.BF → HSF.IF | 0.77 | 0.95 | High School Functions - Building Functions → High School Functions - Interpreting Functions |
| 7.G → HSG.CO | 0.35 | 0.95 | Grade 7 - Geometry → Congruence |
| HSN.RN → HSA.REI | 0.93 | 0.99 | High School Number and Quantity - The Real Number System → High School Algebra - Reasoning with Equations and Inequalities |
| 6.SP → HSS.ID | 0.51 | 1.00 | Grade 6 - Statistics and Probability → High School Statistics and Probability - Interpreting Categorical and Quantitative Data |

Table 4.12: Top association rule for non-mastery (score 0) in each grade

| **Rule** | **S** | **C** | **Rule Description** |
|---|---|---|---|
| 1.G → 1.MD | 0.33 | 0.81 | Grade 1 - Geometry → Grade 1 - Measurement and Data |
| 2.OA → 2.NBT | 0.41 | 0.93 | Grade 2 - Operations and Algebraic Thinking → Grade 2 - Number and Operations in Base Ten |
| 3.NBT → 3.MD | 0.20 | 0.75 | Grade 3 - Number and Operations in Base Ten → Grade 3 - Measurement and Data |
| 4.NBT → 4.OA | 0.48 | 0.88 | Grade 4 - Number and Operations in Base Ten → Grade 4 - Operations and Algebraic Thinking |
| 5.MD → 5.NBT | 0.34 | 0.66 | Grade 5 - Measurement and Data → Grade 5 - Number and Operations in Base Ten |
| 5.NF → 6.NS | 0.12 | 0.87 | Grade 5 - Number and Operations - Fractions → Grade 6 - The Number System |
| 7.G → 7.RP | 0.45 | 0.80 | Grade 7 - Geometry → Grade 7 - Ratios and Proportional Relationships |
| 7.G → 8.G | 0.26 | 0.96 | Grade 7 - Geometry → Grade 8 - Geometry |
| HSA.CED → HSA.REI | 0.55 | 0.86 | High School Algebra - Creating Equations → High School Algebra - Reasoning with Equations and Inequalities |
| HSF.BF → HSF.IF | 0.71 | 0.89 | High School Functions - Building Functions → High School Functions - Interpreting Functions |
| 8.G → HSG.CO | 0.22 | 0.72 | Grade 8 - Geometry → High School Geometry - Congruence |
| HSN.RN → HSA.REI | 0.81 | 0.92 | High School Number and Quantity - The Real Number System → High School Algebra - Reasoning with Equations and Inequalities |
| 6.SP → HSS.ID | 0.51 | 1.00 | Grade 6 - Statistics and Probability → High School Statistics and Probability - Interpreting Categorical and Quantitative Data |

all grades, we present Tables 4.11 and 4.12, which provide the top association rules for mastery and non-mastery across all grades, respectively, with a lower minimum support of 0.1. Specifically, these rules were generated by taking the top rule, the one with high support, from each grade. Analyzing Tables 4.11 and 4.12 several interesting observations can be made:

- There is a clear indication that foundational concepts within the same grade are interdependent, and mastery or non-mastery in one area can affect the learning outcome in another. For instance, in Grade 2, mastery in “Operations and Algebraic Thinking” leads to mastery in “Number and Operations in Base Ten” with a confidence of 0.96 (Table 4.11). Similarly, a lack of mastery in “Operations and Algebraic Thinking” has a high likelihood of resulting in non-mastery in "Number and Operations in Base Ten" with a confidence of 0.93 (Table 4.12).

- ➪ The lower minimum support of 0.1 allows for the identification of rules across different grades, showing the long-term effect of mastering or failing to master certain skills. For example, mastery in "Grade 4 - Number and Operations in Base Ten" has an association with mastery in "Grade 6 - The Number System" with a confidence of 0.75. This association may imply that early proficiency in number operations forms a crucial foundation for understanding the number system in later grades (Table 4.11).

- ➪ Some rules exist both in the mastery and non-mastery tables, emphasizing the critical role of certain skills. One such rule is "Grade 7 - Geometry → Grade 7 - Ratios and Proportional Relationships", suggesting a strong interconnection between understanding geometry and ratios in Grade 7.

- ➪ High School associations are well-represented. High School topics such as Algebra and Functions show up with high support and confidence. For example, the rule "High School Functions - Building Functions → High School Functions - Interpreting Functions" appears in both tables with high confidence, indicating that the ability to build functions is highly indicative of the ability to interpret them, and vice versa.

These findings underline the effectiveness of association rule mining in capturing the relationships between different mathematics skills across grade levels and success statuses, providing invaluable insights for personalized teaching and learning approaches.

# CHAPTER 5

# STUDENT END-OF-UNIT ASSIGNMENT GRADE PREDICTION

In this chapter, we primarily address **RQ3** and **RQ4** through the presentation and discussion of our experimental results. The process of feature engineering, which is crucial to our approach, is detailed in Chapter 5.1. In Chapter 5.2, we delve into a unique type of feature, the graph representation learning. Following this, we outline the machine learning predictive models used in our study in Chapter 5.3. Implementation settings are subsequently clarified in Chapter 5.4 followed by evaluation metrics in Chapter 5.5. Finally, we detail the experimental results of predicting end-of-unit student grades in Chapter 5.6, where we also discuss the significance and implications of our findings.

## 5.1 Feature Engineering

Feature engineering plays a crucial role in machine learning models as it involves selecting and transforming raw data into a format that can be effectively used for training and evaluation. To prepare the data for machine learning models, we performed feature extraction from the given dataset. The feature extraction process involved combining the training and evaluation data and extracting relevant information mainly from four tables: *action_logs*, *assignment_details*, *problem_details*, and *sequence_details*– See Chapter 3. Next, we explain each group of features.

- **Action Log Features.** To retrieve the action logs associated with unit test assignment log IDs, we utilized the assignment relationships table to obtain all the in-unit assignment log IDs corresponding to each unit test assignment log ID– See Table 3.1. Using these in-unit assignment log IDs, we retrieved the relevant action log features. However, it is important to note that even within each in-unit assignment log ID, different problems had varying values for each feature. For example, for the same in-unit assignment log ID, problem ID '28UPV22XPX' had a maximum of 3 attempts, while problem ID 'C9YS98XFX' had a maximum of 1 attempt. On the other hand, machine learning models require only one record for each combination of the unit test assignment log ID and the corresponding end of the unit problem ID. Therefore, it was crucial to find an appropriate approach to aggregate this information for each unit assignment log ID and problem ID group. To resolve this issue, for certain feature types such as *'max attempts'*, *'score viewable'*, and *'continuous score viewable'*, we selected the most frequently occurring value within the group. For *'action'* and *'available core tutoring'* features, we performed one-hot encoding, and the sum of the values was taken for the group. Taken together, the size of Action Log Features ended up being 21.

- **Assignment Detail Features.** Sequence IDs were obtained for all unit test assignment IDs from the assignment details table. This sequence Id is further used to get sequence details features. Additionally, we calculated a new feature called *'% of assignment not finished'* based on the assignment end date. We determined whether an assignment was completed or not by checking if the assignment end date was empty. If the end date was empty, it indicated that the student had not completed the assignment, and we marked those instances as 'not completed.' For each unit-test assignment log ID, we calculated the percentage of uncompleted assignments.

- **Sequence Detail Features.** Based on the sequence ID, we retrieved sequence-related features such as *'sequence folder path levels'* 1, 2, 3, and 4 from the sequence details table. These four features were represented using one-hot encoding, adding 172 dimensions to the feature vector due to the unique values in each of the sequence folder path levels.

- **Problem Detail Features.** For the end-of-unit problems in the training and evaluation sets, we obtained all problem-related features from the problem details table. The feature *'problem skill code'* was split into four parts because each part holds significance according to the CCSS. After one-hot encoding, this added 133 dimensions to the feature vector. Additionally, the *'problem type'* feature was represented using one-hot encoding, resulting in 10 dimensions. We represented the *'problem skill description'* feature by 32-dimensional BERT PCA embeddings, while the BERT PCA embeddings for the *'problem text'* were already provided in the data. Other problem features, such as *'problem contains image'*, *'problem contains video'* and *'problem contains equation'*, were already in binary format and used as they were. Additionally, we introduced a new feature called *'problem multipart ID frequency'*, taking into account the problem multipart ID present in the dataset. This feature captures the number of occurrences of a specific problem multipart ID within the dataset. Taken together, the size of Problem Detail Features ended up being 210.

Table 5.1: Description of features and their dimensions

| **Feature Category** | **Brief Description** | **Dim** |
|---|---|---|
| Action Log | These include features related to action logs associated with unit test assignment log IDs. Information for these features is retrieved and aggregated from various sources. This includes different approaches for different types of features like 'max attempts', 'score viewable', 'continuous score viewable', 'action', and 'available core tutoring'. | 21 |
| Assignment Detail | These include features related to sequence IDs and a calculated feature known as '% of assignment not finished' based on the assignment end date. | 1 |
| Sequence Detail | These features are based on the sequence ID and include 'sequence folder path levels' 1, 2, 3, and 4. These features are represented using one-hot encoding. | 172 |
| Problem Detail | These include various problem-related features obtained from the problem details table. Features are processed differently based on their type, such as 'problem skill code', 'problem type', 'problem skill description', 'problem text', and others. A new feature, 'problem multipart ID frequency' was also introduced. | 210 |
| | **Total:** | **404** |

All other features, like explanation details and hint details, which were mentioned in the Chapter 3 but not included in this feature extraction explanation, did not contribute significantly to improving the performance of the machine learning models. Therefore, we excluded them from the feature set described here. In summary, the feature extraction process involved combining data from multiple tables, handling varying values within groups, performing one-hot encoding, and utilizing existing embeddings. This resulted in expanded data with additional dimensions, enhancing the representation of the data for machine learning models. For convenience, Table 5.1 includes a brief description of each feature category and its dimension.

## 5.2 Graph Representation Learning

### 5.2.1 Graph Structure.

As an additional step towards understanding and modeling student behavior within the ASSISTments learning platform, we delved into graph representation learning. Graph representation learning involves capturing relationships within a graph to learn informative node and edge representations [60]. We hypothesize that the structural features hidden in the relationships between different entities in the dataset could help predict the outcome of the end-of-unit tests. To validate this hypothesis, first, we identified 5 entities in the dataset as nodes in the graph: 'student', 'teacher', 'class', 'problem', and 'sequence'. Also, we used 4 types of connection (edge types) that are sufficient to represent the relations between the entities in the dataset: 'teacher-class', 'class-student', 'student-problem', and 'problem-sequence'. Other connections, such as 'student-teacher', can be encoded by a combination of 'student-class' and 'class-teacher' edges. This helped us avoid high edge density in the graph while retaining the important structural information. Figure 5.1 shows the structure of the final constructed graph.

### 5.2.2 Graph Construction.

To populate the graph, we first added nodes for students that took the end-of-unit tests. For these students, we used the assignment_relationships table to find all of the related in-unit assignments that they were assigned. The information about the classes those students took (in-unit as well as end-of-unit) and their teachers is also available

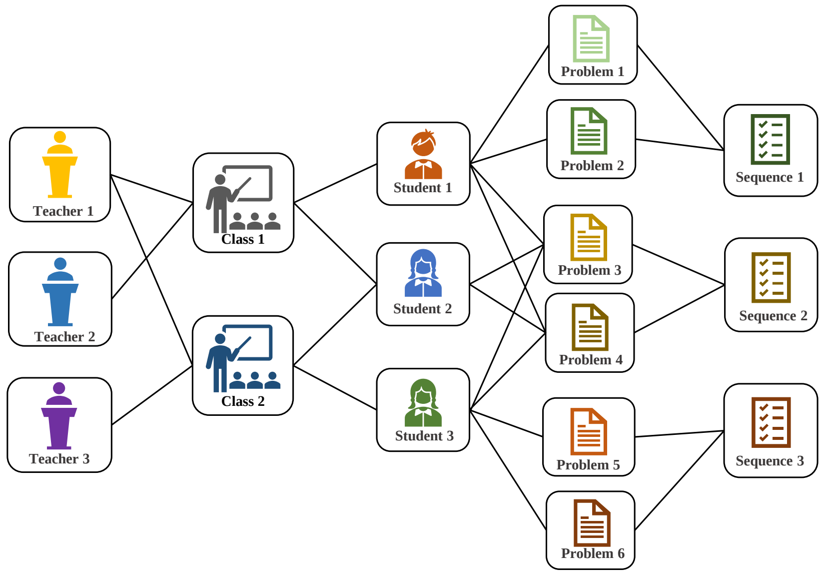


Fig. 5.1: The constructed graph between different entities in the dataset

in the assignment_details table. Therefore, we added nodes for classes and teachers and connected students to their classes and classes to teachers who taught them. We then added nodes for the end-of-unit problems and created edges between the problems and the students who worked on them. Then, again using the assignment_relationships table, we found the relevant action log records to each end-of-unit assignment where we could find the problems that the students completed within the unit. Subsequently, we created edges between students and the relevant problems they performed an action on during the unit. For each problem (end-of-unit or within the unit), we also found their corresponding sequence and created an edge between problems and their sequence. Table 5.2 shows the basic properties of the constructed graph.

### 5.2.3 Graph Representation Learning.

We utilized a random-walk-based representation learning algorithm named node2vec [17] to map the nodes in the graph to an embedding space. node2Vec is a popular algorithm designed to capture the structural and community properties of nodes in a graph by generating low-dimensional vector representations, or embeddings, for each node. node2Vec builds on the concept of random walks within a graph [61]. It explores the idea that nodes

Table 5.2: Properties of the constructed graph.

| Attribute | Value |
|---|---|
| Type | Heterogeneous |
| # Node types | 5 |
| # Students | 34652 |
| # Problems | 59109 |
| # Sequences | 5766 |
| # Teachers | 2024 |
| # Classes | 3055 |
| # Total nodes | 104606 |
| # Edges | 5527865 |
| Density | 0.001 |

that are close to each other in the graph tend to have similar roles and functions. The algorithm achieves this by sampling random walks of a specified length from each node in the graph. These random walks capture the local neighborhood information around each node. By treating each random walk as a sequence of nodes, node2Vec uses a modified Skip-gram model, a popular method from the field of natural language processing [62], to learn embeddings. The Skip-gram model is trained to predict the likelihood of encountering a node in the random walk based on its neighboring nodes. After applying node2vec, we used the learned embedding for the end-of-unit problem to enrich the representation even further. We also ran experiments to investigate the effectiveness of embeddings in grade prediction without the use of hand-crafted features explained in Chapter 5.1. Our experiments show that graph representation learning is beneficial in the grade prediction task. We detail our experimental results in Chapter 5.6.

### 5.3 Predictive Models

For the end-of-unit grade prediction, we employed an assortment of models, including Random Forest, Gradient Boosting, XGBoost, LightGBM, ExtraTrees, and a Mean Ensemble of the previous five models. The rationale behind this selection is multi-fold:

- **Robustness and Flexibility:** All these models, namely Random Forest, Gradient

Boosting, XGBoost, and ExtraTrees, are known for their robustness and flexibility [63, 64]. They can handle different types of data and are less prone to overfitting compared to other machine learning models.

- **Performance:** XGBoost and LightGBM are gradient boosting frameworks that have proven to be very efficient and effective in a wide range of regression and classification tasks [65, 66]. They have also shown superior performance in numerous machine learning competitions.

- **Handling High-Dimensional Spaces:** Random Forest and ExtraTrees are particularly effective in high-dimensional spaces and can model complex interactions between features [67, 68].

- **Ensemble Learning:** Ensemble learning is a powerful way to improve model performance by combining several base models, which can reduce variance (bagging techniques like Random Forest and ExtraTrees) or bias (boosting techniques like Gradient Boosting, XGBoost, and LightGBM) [69]. The Mean Ensemble model was used to capitalize on this benefit by averaging the predictions of the individual models, thereby further improving the overall performance and robustness of the prediction.

In summary, these models were selected due to their ability to handle various types of data, superior performance records, flexibility, and the advantage of ensemble learning which is conducive to achieving more accurate and robust predictions. Next, we present a brief technical description of each model.

### 5.3.1 Random Forest

Random Forest is a machine learning algorithm that combines multiple decision trees to make predictions. It is an ensemble learning method that aggregates the results of individual trees to produce a final prediction. Each tree in the forest is trained on a random subset of the data and features, reducing overfitting and improving generalization. Formally, given a dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, Random Forest builds $T$ decision trees $\{h_t(x)\}_{t=1}^{T}$ using bootstrapped subsets of $D$. The final prediction is obtained by averaging the individual tree predictions:

$$H(x) = \frac{1}{T}\sum_{t=1}^{T} h_t(x) \tag{5.1}$$

### 5.3.2 Gradient Boosting

Gradient Boosting is a machine learning algorithm that builds an ensemble of weak prediction models, typically decision trees, in a sequential manner. It works by iteratively adding models to correct the errors made by previous models. Each new model is trained on the residuals of the previous models, optimizing the overall prediction. Formally, given a dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, gradient boosting builds an ensemble of weak learners $\{h_t(x)\}_{t=1}^{T}$ by minimizing the loss function $L(y, F(x))$:

$$F(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{5.2}$$

where $\alpha_t$ is the step size at iteration $t$.

### 5.3.3 XGBoost

XGBoost (eXtreme Gradient Boosting) is an ensemble method that combines multiple weak prediction models, typically decision trees, to create a strong predictive model. XGBoost employs a unique gradient boosting framework that optimizes a loss function by iteratively adding new models to correct the errors made by the previous models. Formally, given a dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, XGBoost builds an ensemble of $T$ decision trees $\{h_t(x)\}_{t=1}^{T}$ by minimizing the regularized loss function $L(y, F(x)) + \Omega(h_t)$:

$$F(x) = \sum_{t=1}^{T} h_t(x) \tag{5.3}$$

where $\Omega(h_t)$ is the regularization term that controls the complexity of the decision trees.

### 5.3.4 LightGBM

LightGBM (Light Gradient Boosting Machine) is a gradient boosting framework that uses tree-based learning algorithms. It employs a novel technique called Gradient-based One-Side Sampling (GOSS) to select the most informative instances for training, resulting in faster convergence and reduced memory usage. Given a dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$ consisting of $n$ observations, LGBM builds an ensemble of decision trees $\{h_t(x)\}_{t=1}^{T}$ using a gradient-based boosting framework. The final prediction for a new observation $x$ is obtained by summing the individual tree predictions, weighted by the learning rate ($\eta$):

$$H(x) = \eta \sum_{t=1}^{T} h_t(x)$$

where $\eta$ is the learning rate that scales the contribution of each individual tree prediction in the final ensemble prediction.

### 5.3.5 ExtraTrees

ET, or Extra Trees, is an ensemble learning algorithm that is similar to Random Forests. It builds multiple decision trees using bootstrapped subsets of the training data. However, unlike Random Forests, ET selects random splits for each feature and does not perform feature-specific threshold optimization. The final prediction is made by averaging the predictions of all the individual trees. Formally, similar to Random Forest, Ensemble Trees (ET) is a machine learning algorithm that builds an ensemble of decision trees using bootstrapped subsets of a given dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$. The final prediction in ET is obtained by averaging the individual tree predictions, as shown in the following formula:

$$H(x) = \frac{1}{T} \sum_{t=1}^{T} h_t(x) \tag{5.4}$$

### 5.3.6 Mean Ensemble

The mean ensemble model combines predictions from the five classifiers, including Random Forest (RF), XGBoost (XGB), LightGBM (LGBM), Extra Trees (ET), and Gradient

Boosted Classifier (GBC), by averaging their predictions. By integrating the strengths of each classifier and mitigating biases or weaknesses, this ensemble approach enhances generalization performance and accuracy. The diverse modeling techniques of these classifiers contribute to the ensemble's ability to capture a wide range of patterns and make more robust predictions. Overall, the mean ensemble strategy serves as a powerful tool in machine learning for achieving improved predictive performance.

### 5.4 Implementation Settings

After extracting all the features as explained in Chapter 5.1, we standardized both training and evaluation data using the StandardScaler package from scikit-learn [70]. Standardization helps to bring the features to a common scale, enabling fair comparisons and preventing features with larger magnitudes from dominating the model's learning process. For tuning the hyperparameters of each predictive model mentioned in Chapter 5.3, we utilized the RandomizedSearchCV from scikit-learn. We performed 10-fold cross-validation while tuning the hyperparameters. Additionally, we used the scikit-learn package to implement the Random Forest, Gradient Boosting, and Extra Trees methods. For the XGBoost model, we employed the XGBoost Python package [71], while we used the lightgbm package [72] for implementing the LGBM model. We made use of the node2vec package[1] for the node2vec implementation described in Chapter 5.2. The hyperparameters used in learning node embeddings with node2vec are as follows: embedding dimension: 32, number of walks: 100, walk length: 10, and window size: 10. Ultimately, we implemented all the evaluation metrics mentioned in Chapter 5.5 using the scikit-learn package.

### 5.5 Evaluation Metrics

We utilized different evaluation metrics to measure the performance of our predictive models. These metrics allowed us to quantitatively assess the effectiveness and accuracy of our models in making predictions. Let TP, FP, TN, and FN are the number of true positive,

[1] https://pypi.org/project/node2vec/

false positive, true negative, and false negative samples, respectively. The following are the definitions of the metrics used for evaluation:

*Accuracy* is a metric that measures the overall correctness of a model's predictions. It calculates the ratio of the number of correct predictions to the total number of predictions made.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

*Precision* is a metric that quantifies the model's ability to correctly identify positive instances out of the total instances it predicted as positive.

$$Precision = \frac{TP}{TP + FP}$$

*Recall* measures the model's ability to identify all positive instances correctly.

$$Recall = \frac{TP}{TP + FN}$$

*F1-score* is the harmonic mean of precision and recall. It provides a single metric that combines both precision and recall, giving a balanced measure of a model's performance.

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

*AUC* measures the ability of the model to correctly classify positive and negative samples across different thresholds. It represents a probability curve that quantifies the model's separability between classes. A higher AUC indicates better predictive capability, as it signifies a stronger ability to accurately classify samples as belonging to their respective classes.

### 5.6 Experimental Results

Table 5.3 summarizes the results of our extensive experiments. We define three experimental settings to illustrate the effectiveness of graph embedding techniques in the

Table 5.3: Performance of developed ML models across three settings and for different evaluation metrics. In each column and for each setting, the bold value indicates the best model according to that metric. The underlined value indicates the best across all settings.

| Setting | | Model | Accuracy | Precision | Recall | F1 score | AUC |
|---|---|---|---|---|---|---|---|
| (I) | Hand-crafted features (Chapter 5.1) | XGBoost | 0.72238 | **0.74441** | 0.80658 | 0.77425 | 0.78295 |
| | | Random Forest | 0.71402 | 0.73045 | 0.81694 | 0.77128 | 0.77005 |
| | | Extra Trees | 0.69865 | 0.70906 | <u>**0.83001**</u> | 0.76479 | 0.74925 |
| | | LGBM | 0.72370 | 0.74361 | 0.81177 | 0.77620 | **0.78473** |
| | | Gradient Boosting | 0.72257 | 0.74296 | 0.81030 | 0.77517 | 0.78245 |
| | | Mean Ensemble | **0.72437** | 0.74043 | 0.82075 | <u>**0.77853**</u> | 0.78383 |
| (II) | Student + Problem embedding (Chapter 5.2) | XGBoost | 0.68305 | **0.72894** | 0.73711 | 0.73300 | 0.73251 |
| | | Random Forest | **0.69458** | 0.72626 | 0.77444 | **0.74958** | **0.74286** |
| | | Extra Trees | 0.69094 | 0.72168 | **0.77543** | 0.74759 | 0.74115 |
| | | LGBM | 0.68579 | 0.72203 | 0.76039 | 0.74071 | 0.73338 |
| | | Gradient Boosting | 0.69068 | 0.72525 | 0.76621 | 0.74517 | 0.73982 |
| | | Mean Ensemble | 0.69165 | 0.72763 | 0.76329 | 0.74504 | 0.74174 |
| (III) | Hand-crafted features (Chapter 5.1) + Problem embedding (Chapter 5.2) | XGBoost | 0.70857 | 0.74560 | 0.76843 | 0.75685 | 0.76520 |
| | | Random Forest | 0.71218 | 0.73662 | 0.79750 | 0.76585 | 0.77084 |
| | | Extra Trees | 0.71737 | 0.74689 | 0.78831 | 0.76704 | 0.77279 |
| | | LGBM | 0.71967 | <u>**0.74990**</u> | 0.78780 | 0.76838 | 0.78145 |
| | | Gradient Boosting | 0.71507 | 0.74887 | 0.77825 | 0.76328 | 0.77443 |
| | | Mean Ensemble | <u>**0.72784**</u> | 0.74979 | **0.80880** | **0.77818** | <u>**0.78977**</u> |

end-of-unit grade prediction task.

- **Setting (I):** This setting only uses hand-crafted features, detailed in Chapter 5.1, as input to the machine learning models. These features are derived from the raw data without the application of graph embedding techniques. The aim here is to evaluate the performance of models based on explicit feature engineering.

- **Setting (II):** This setting uses a combination of the end-of-unit problem embeddings and student embeddings as inputs to the models. Embedding techniques are used here to represent the problems and students in a high-dimensional space. The purpose of this setting is to investigate the effectiveness of embedding methods in representing students and problems.

- **Setting (III):** In this setting, the models are trained on data that combines both hand-crafted features and problem embeddings. This setting is designed to study whether the combination of hand-crafted features and embeddings can improve the performance of the models in predicting students' scores.

Based on the results presented in Table 5.3, we make the following observations.

- In setting (I), which uses only hand-crafted features, the Mean Ensemble model performs best in terms of accuracy and F1 score. Notably, Extra Trees model gives the best recall, indicating that it is the best at identifying true positive cases.
- For setting (II), where we use the embeddings of the student and end-of-unit problem, the Random Forest model seems to outperform the others, providing the highest accuracy and F1 score. However, the Extra Trees model provides the highest recall, similar to setting (I).
- In setting (III), where both hand-crafted features and problem embeddings are used, the Mean Ensemble model outperforms all other models in terms of accuracy, F1 score, and AUC. This shows the benefits of combining various models' predictions.
- In setting (I), all models perform very closely to each other in terms of AUC, suggesting that hand-crafted features can provide a consistent baseline for different models.
- In setting (II), despite only using low-dimensional features and lacking information related to students' actions within the units, a relatively high AUC is achieved. This indicates the power of embedding techniques in capturing complex structural relationships in the data without the need for complicated hand-crafted features, which may not be obtained readily.
- In setting (III), the results not only confirmed the effectiveness of graph embedding but also showed that combining these embeddings with hand-crafted features can further enhance prediction performance. This setting achieved the best results in terms of accuracy and precision among all settings.
- The Mean Ensemble model consistently demonstrates strong performance across all settings, often achieving the highest or near-highest values for several metrics. This suggests that the ensemble approach, which leverages the strengths of multiple models, can effectively enhance the prediction performance.

- ➪ The Extra Trees model shows particular strength in recall across both settings I and II, indicating its ability to correctly identify positive instances in the dataset.
- ➪ XGBoost performs consistently well, particularly in terms of precision in settings I and III. This highlights the model's ability to limit the number of false positives in its predictions.
- ➪ LightGBM and Gradient Boosting have relatively stable performance across different settings, showing their robustness in different feature spaces.
- ➪ The performance of Random Forest is notably high in setting II, where only embeddings are used, demonstrating its ability to handle high-dimensional data.
- ➪ Overall, we observe that combining hand-crafted features with problem embeddings (setting III) provides the highest performance across most metrics, underlining the importance of integrating domain-specific features with learned embeddings for educational grade prediction. Notably, the AUC, the metric used in EDM Cup 2023 Kaggle Competition, is the highest in this setting compared to the other two.

**Feature Importance.** Figure 5.2 demonstrates the relative importance of different hand-crafted features in predicting the student grade in setting (I). We make the following observations based on the results presented in this figure.

- ➪ The most important feature appears to be the end-of-unit problem text, which is provided as a 32-dimensional embedding vector in the dataset.
- ➪ Actions that students took within the unit, such as tutoring requests and responses to questions, are shown to significantly affect the student grades, reinforcing our analysis in Chapter 4.1.
- ➪ The problem skill description and problem skill code also feature among the top 5 most important features, reflecting the relationship between students' grades and the skills they are being tested on. This point underscores the importance of our analysis in Chapter 4.2.

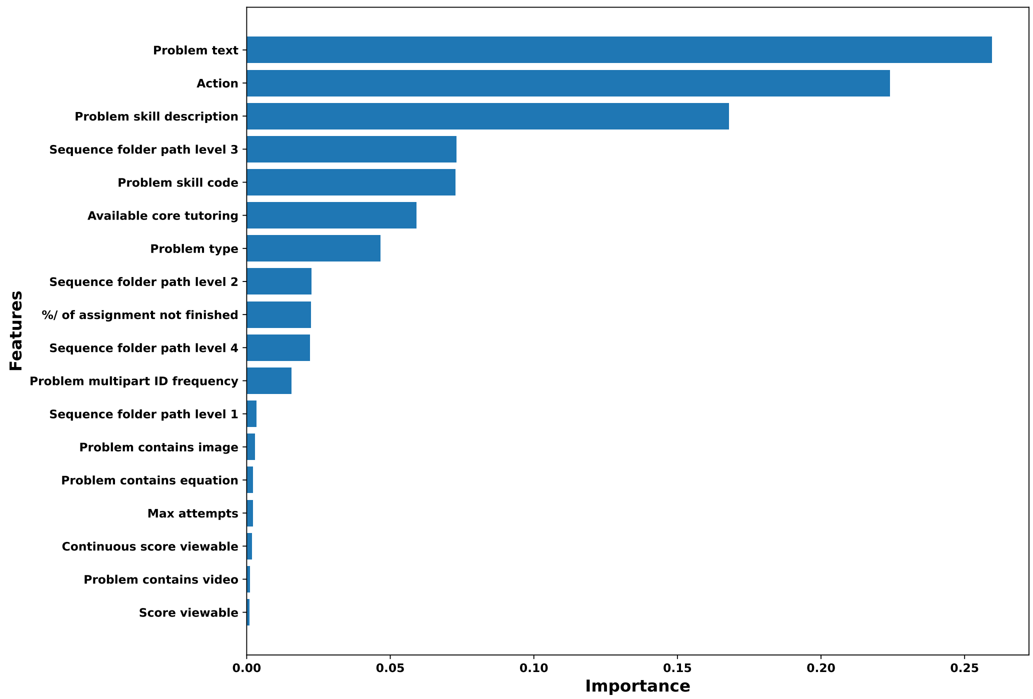


Fig. 5.2: Feature importance of hand-crafted features explained in Chapter 5.1

- The sequence folder path level 3, which describes what unit the sequence is part of, is among the top 5 important features. More descriptions for other sequence path levels can be found on the dataset website [49].

- The importance of available core tutoring types within the unit in predicting the student's grade is worth noting, implying that instructional methods can be indicative of student performance.

CHAPTER 6

# CONCLUSION

In this chapter, we begin by presenting a summary of our study in Chapter 6.1. Following this, we will delve into the limitations of our analytical framework and the dataset in Chapter 6.2. We will conclude the chapter by highlighting several promising avenues for future research in Chapter 6.3.

## 6.1 Summary

This thesis presented a machine learning approach to predict end-of-unit grades in an intelligent tutoring system, specifically focusing on the ASSISTments learning platform. The study explored the use of both hand-crafted features and graph representation learning to improve prediction accuracy. The thesis began by introducing the motivation behind the study, highlighting the importance of predicting student performance in educational systems to provide personalized interventions and support. It also emphasized the potential of intelligent tutoring systems to enhance student learning outcomes. Next, we discussed the dataset used in the study, which contained detailed information about student actions, problem features, and performance outcomes. We explained how various features, such as problem text, student actions, problem skills, and sequence information, were extracted from the dataset to create hand-crafted features for prediction. The thesis then delved into graph representation learning, where relationships between entities in the dataset were modeled using a graph structure. Five entities, including students, teachers, classes, problems, and sequences, were represented as nodes in the graph, and connections between them were defined as edges. The constructed graph provided valuable structural information for predicting end-of-unit grades. After constructing the graph, a random walk-based representation learning algorithm called node2vec was employed to map the nodes to an embedding space. The node2vec algorithm captured the local neighborhood information of each node and generated low-dimensional vector representations. The learned embeddings, along with the hand-crafted features, enriched the representation of end-of-unit problems and enhanced the prediction accuracy. To evaluate the predictive models, we employed an assortment of machine learning algorithms, including Random Forest, Gradient Boosting, XGBoost, LightGBM, Extra Trees, and a Mean Ensemble of these models. Each model was evaluated based on various metrics such as accuracy, precision, recall, F1 score, and AUC. The thesis defined three experimental settings to analyze the effectiveness of hand-crafted features and graph embeddings. The results of the experiments showed that the Mean Ensemble model consistently performed well across different settings, often achieving the highest or near-highest values for multiple metrics. The combination of hand-crafted features and problem embeddings (Setting III) yielded the best performance, indicating the importance of integrating domain-specific features with learned embeddings. The Extra Trees model demonstrated strength in recall, XGBoost exhibited stability and precision, and Random Forest performed effectively in handling high-dimensional data. The thesis also included a feature importance analysis, which highlighted the significance of different hand-crafted features in predicting student grades. Features related to problem text, student actions within the unit, problem skills, and sequence information contributed significantly to the prediction performance.

In conclusion, the thesis provided an in-depth analysis of student behavior in the ASSISTments learning platform. We discovered meaningful and informative patterns about tutoring requests and CCSS-related skill mastery/non-mastery. Also, the thesis presented a comprehensive machine learning approach to predict end-of-unit grades in the ASSISTments learning platform. By combining hand-crafted features and graph embeddings, the study achieved improved predictive performance. The findings emphasized the power of ensemble learning and the importance of integrating domain-specific knowledge with learned representations. The results had practical implications for the development of intelligent tutoring systems and personalized learning interventions to support student success in educational settings.

### 6.2 Limitations

While the machine learning approach presented in this thesis has shown promising results in predicting end-of-unit grades in the ASSISTments learning platform, it is important to acknowledge several limitations that should be considered when interpreting the findings and implications of this research.

- ✘ **End-of-unit Test Administration Clarification**: The administration method of end-of-unit tests remains ambiguous. For example, it's uncertain whether these tests have an associated action log. Obtaining further details about these end-of-unit tests could potentially enhance our analysis and score prediction capability.

- ✘ **Concealed Problem Text**: As highlighted in Chapter 5.6 (Figure 5.2), the problem's textual description is a key feature in predicting end-of-unit problem scores. Regrettably, we only had access to the BERT embedding of the text; the actual description remained undisclosed. We posit that access to this textual information could provide deeper insights into the nature of the problems, potentially leading to a more comprehensive understanding of students' behavior and the ASSISTments platform.

- ✘ **Generalizability of Findings**: The findings and results of this study are based on a specific dataset from the ASSISTments learning platform. The generalizability of the predictive models and their performance in other educational platforms or contexts may vary. Different platforms may have unique features, student populations, or educational settings that could impact the effectiveness and applicability of the models.

- ✘ **Data Availability and Completeness**: The availability and completeness of the data can affect the accuracy and robustness of the predictive models. In this study, the analysis relies on the data provided by the ASSISTments learning platform, which may have limitations or missing data points. The presence of missing data or incomplete records could introduce biases or impact the model's ability to capture certain patterns or relationships.

- ✘ **Limited Feature Space**: The feature space used in the predictive models is based on the available data and domain-specific features. While efforts have been made to extract relevant features, there may be other unexplored features or contextual factors that could potentially improve the prediction accuracy. Incorporating a wider range of features, such as additional demographic or socio-economic information, may provide a more comprehensive understanding of student performance.

- ✘ **Potential Bias and Fairness Issues**: Predictive models in educational contexts need to be carefully evaluated for potential biases and fairness issues. The models' predictions and outcomes may be influenced by factors such as student demographics, prior academic performance, or socio-economic backgrounds. It is important to assess the models' performance across different subgroups to ensure they are not perpetuating existing inequalities or disadvantaging certain student populations.

- ✘ **Causality and Interpretability**: Predictive models provide associations and correlations between variables but may not establish causal relationships. While the models can accurately predict end-of-unit grades, they may not fully explain the underlying factors or mechanisms driving student performance. Further research is needed to uncover the causal links and provide a deeper understanding of the factors that contribute to student success.

- ✘ **Ethical Considerations**: The use of predictive models in education raises ethical considerations related to student privacy, informed consent, and data protection. It is crucial to adhere to ethical guidelines and regulations to ensure the responsible and ethical use of student data. Safeguards should be implemented to protect student privacy and ensure data security throughout the data collection, analysis, and storage processes.

- ✘ **Human Factors and Teacher Influence**: The predictive models in this study focus primarily on student-level factors and do not explicitly consider the influence of teachers or instructional practices on student outcomes. Teachers play a crucial role in shaping student learning experiences, and their impact on student performance should be taken into account for a comprehensive understanding of grade prediction.

By acknowledging these limitations, researchers and educators can better interpret the findings of this study and recognize areas where further research and improvements are needed. These limitations provide opportunities for future research to address the gaps and refine the predictive models for more accurate and contextually relevant grade predictions.

## 6.3 Future Directions

While the presented machine learning approach for predicting end-of-unit grades in the ASSISTments learning platform has provided promising results, there are several potential future directions that could further enhance the predictive performance and extend the research in this domain.

- ✰ **Integration of Additional Data Sources**: Exploring the integration of additional data sources, such as demographic information, socio-economic factors, or prior academic performance, could provide a more comprehensive understanding of student behavior and improve the accuracy of grade predictions. By incorporating a wider range of features, the models may capture additional patterns and relationships that contribute to student success.

- ✰ **Temporal Analysis**: Conducting a more detailed temporal analysis could offer insights into the progression of student performance over time. Examining how student actions and problem-solving skills evolve throughout a unit or across multiple units could provide valuable information for predicting end-of-unit grades. Temporal modeling techniques, such as recurrent neural networks or attention mechanisms, could be explored to capture the sequential nature of student interactions.

- ✰ **Dynamic Adaptation**: Investigating the potential for dynamic adaptation in the prediction models could further enhance their effectiveness. By continuously updating and adapting the models based on real-time student data, the models can be fine-tuned to account for individual variations and changes in student behavior. This adaptive approach would enable more personalized and timely interventions to support student learning and improve grade prediction accuracy.

- ✰ **Interpretability and Explainability**: Enhancing the interpretability and explainability of the predictive models is crucial for building trust and understanding in the educational context. Developing techniques to explain the model's predictions and provide transparent insights into the features and relationships influencing the predictions would be valuable. This could facilitate effective communication between educators, students, and other stakeholders, leading to more informed decision-making.

- ✰ **Transferability and Generalizability**: Assessing the transferability and generalizability of the developed models to other educational platforms and contexts would be valuable. Validating the models on diverse datasets from different educational systems or domains would provide insights into their robustness and applicability beyond the specific ASSISTments learning platform. This would contribute to the broader field of educational data mining and help identify best practices for grade prediction in various settings.

- ✰ **Intervention Strategies**: Moving beyond grade prediction, exploring the potential for developing intervention strategies based on the predictive models could be an interesting avenue for future research. Leveraging the predictions to identify students at risk and designing targeted interventions or adaptive learning paths could have a significant impact on student outcomes. Evaluating the effectiveness of these interventions and their impact on student performance would be an important next step.

- ✰ **Ethical Considerations**: As with any predictive modeling in education, ethical considerations should be a priority. Investigating potential biases, fairness, and equity issues in the predictive models should be addressed. Ensuring that the models are sensitive to individual differences, avoid perpetuating inequalities, and promote inclusive educational practices is crucial.

By exploring these future directions, researchers can further advance the field of predictive analytics in intelligent tutoring systems, improve the accuracy of grade predictions, and contribute to the development of effective personalized learning interventions to support student success.

# REFERENCES


[1] A. Bilyalova, D. Salimova, and T. Zelenina, “Digital transformation in education,” in *Integrated Science in Digital Age: ICIS 2019*. Springer, 2020, pp. 265–276.

[2] Z.-Y. Liu, N. Lomovtseva, and E. Korobeynikova, “Online learning platforms: Reconstructing modern higher education,” *International Journal of Emerging Technologies in Learning (iJET)*, vol. 15, no. 13, pp. 4–21, 2020.

[3] T. Dillahunt, Z. Wang, and S. D. Teasley, “Democratizing higher education: Exploring mooc use among those who cannot afford a formal education,” *International Review of Research in Open and Distributed Learning*, vol. 15, no. 5, pp. 177–196, 2014.

[4] T. J. Blayone, R. vanOostveen, W. Barber, M. DiGiuseppe, and E. Childs, “Democratizing digital learning: theorizing the fully online learning community model,” *International Journal of Educational Technology in Higher Education*, vol. 14, pp. 1–16, 2017.

[5] Z. Akyol and D. R. Garrison, “Understanding cognitive presence in an online and blended community of inquiry: Assessing outcomes and processes for deep approaches to learning,” *British Journal of Educational Technology*, vol. 42, no. 2, pp. 233–250, 2011.

[6] H. Abuhassna, W. M. Al-Rahmi, N. Yahya, M. A. Z. M. Zakaria, A. B. M. Kosnin, and M. Darwish, “Development of a new model on utilizing online learning platforms to improve students’ academic achievements and satisfaction,” *International Journal of Educational Technology in Higher Education*, vol. 17, pp. 1–23, 2020.

[7] K. Livingston and R. Condie, “The impact of an online learning program on teaching and learning strategies,” *Theory into Practice*, vol. 45, no. 2, pp. 150–158, 2006.

[8] K. A. Meyer, "Student engagement in online learning: What works and why," *ASHE higher education report*, vol. 40, no. 6, pp. 1–114, 2014.

[9] M. Akkus, "The common core state standards for mathematics." *International Journal of Research in Education and Science*, vol. 2, no. 1, pp. 49–54, 2016.

[10] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *ACM SIGMOD Record*, vol. 22, no. 2. ACM, 1993, pp. 207–216.

[11] T. A. Kumbhare and S. V. Chobe, "An overview of association rule mining algorithms," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 1, pp. 927–930, 2014.

[12] R. F. Kizilcec, C. Piech, and E. Schneider, "Deconstructing disengagement: analyzing learner subpopulations in massive open online courses," in *Proceedings of the third international conference on learning analytics and knowledge*. ACM, 2013, pp. 170–179.

[13] L. Breslow, D. E. Pritchard, J. DeBoer, G. S. Stump, A. D. Ho, and D. T. Seaton, "Studying learning in the worldwide classroom: Research into edx's first mooc," *Research & Practice in Assessment*, vol. 8, pp. 13–25, 2013.

[14] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart, "Predicting mooc dropout over weeks using machine learning methods," in *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*. ACM, 2014, pp. 60–65.

[15] H. Karimi, T. Derr, J. Huang, and J. Tang, "Online academic course performance prediction using relational graph convolutional neural network." *International Educational Data Mining Society*, 2020.

[16] M. Li, X. Wang, Y. Wang, Y. Chen, and Y. Chen, "Study-gnn: A novel pipeline for student performance prediction based on multi-topology graph neural networks," *Sustainability*, vol. 14, no. 13, p. 7965, 2022.

[17] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[18] F. Chen, Y.-C. Wang, B. Wang, and C.-C. J. Kuo, “Graph representation learning: a survey,” *APSIPA Transactions on Signal and Information Processing*, vol. 9, p. e15, 2020.

[19] H. Karimi, T. Derr, K. T. Torphy, K. A. Frank, and J. Tang, “A roadmap for incorporating online social media in educational research,” *Teachers College Record*, vol. 121, no. 14, pp. 1–24, 2019. [Online]. Available: https://doi.org/10.1177/016146811912101412

[20] H. Karimi, J. Tang, X. Weiss, and J. Huang, “Automatic identification of teachers in social media using positive unlabeled learning,” in *2021 IEEE International Conference on Big Data (Big Data)*, 2021, pp. 643–652.

[21] S. Solanki, K. Kheiri, M. A. Tsugawa, H. Karimi *et al.*, “Leveraging social media analytics in engineering education research,” in *2023 ASEE Annual Conference & Exposition*. Baltimore , Maryland: ASEE Conferences, June 2023, https://peer.asee.org/43472.

[22] S. Farokhi, A. Yaramala, J. Huang, M. F. Khan, X. Qi, and H. Karimi, “Enhancing the performance of automated grade prediction in mooc using graph representation learning,” in *2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2023.

[23] J. K. Olsen, D. M. Belenky, V. Aleven, and N. Rummel, “Intelligent tutoring systems for collaborative learning: Enhancements to authoring tools,” in *Artificial Intelligence in Education: 16th International Conference, AIED 2013, Memphis, TN, USA, July 9-13, 2013. Proceedings 16*. Springer, 2013, pp. 900–903.

[24] A. C. Graesser, M. W. Conley, and A. Olney, "Intelligent tutoring systems." *APA educational psychology handbook, Vol 3: Application to learning and teaching.*, pp. 451–473, 2012.

[25] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier, "Cognitive tutors: Lessons learned," *The journal of the learning sciences*, vol. 4, no. 2, pp. 167–207, 1995.

[26] K. Seo, J. Tang, I. Roll, S. Fels, and D. Yoon, "The impact of artificial intelligence on learner–instructor interaction in online learning," *International journal of educational technology in higher education*, vol. 18, pp. 1–23, 2021.

[27] M. Utterberg Modén, M. Tallvid, J. Lundin, and B. Lindström, "Intelligent tutoring systems: Why teachers abandoned a technology aimed at automating teaching processes," 2021.

[28] L. V. Morris, C. Finnegan, and S.-S. Wu, "Tracking student behavior, persistence, and achievement in online courses," *The Internet and Higher Education*, vol. 8, no. 3, pp. 221–231, 2005.

[29] G. Borracci, E. Gauthier, J. Jennings, K. Sale, and K. Muldner, "The effect of assistance on learning and affect in an algebra tutor," *Journal of Educational Computing Research*, vol. 57, no. 8, pp. 2032–2052, 2020.

[30] P. I. Pavlik, H. Cen, and K. R. Koedinger, "Performance factors analysis—a new alternative to knowledge tracing," *Online Submission*, 2009.

[31] C. Romero and S. Ventura, "Educational data mining and learning analytics: An updated survey," *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, vol. 10, no. 3, p. e1355, 2020.

[32] D. Gonda, V. Ďuriš, G. Pavlovičová, and A. Tirpáková, "Analysis of factors influencing students' access to mathematics education in the form of mooc," *Mathematics*, vol. 8, no. 8, p. 1229, 2020.

[33] B. S. Bloom, "The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring," *Educational researcher*, vol. 13, no. 6, pp. 4–16, 1984.

[34] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," in *User modeling and user-adapted interaction*, vol. 4, no. 4. Springer, 1995, pp. 253–278.

[35] L. S. Vygotsky, *Mind in society: The development of higher psychological processes*. Harvard university press, 1978.

[36] D. T. Willingham, "Critical thinking: Why is it so hard to teach?" *American Educator*, vol. 31, no. 2, pp. 8–19, 2007.

[37] M. Feng, N. T. Heffernan, and K. R. Koedinger, "Learning trajectories for the common core state standards for mathematics," in *International Conference on Intelligent Tutoring Systems*. Springer, 2014, pp. 322–328.

[38] N. Heffernan and C. Heffernan, "The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching," in *International Conference on Intelligent Tutoring Systems*. Springer, 2014, pp. 494–504.

[39] A. Porter, J. McMaken, J. Hwang, and R. Yang, "Common core standards the new us intended curriculum," *Educational Researcher*, vol. 40, no. 3, pp. 103–116, 2011.

[40] K. R. Koedinger, J. Stamper, B. Leber, and A. Skogsholm, "Mining the gap: using learning analytics to discover missing prerequisites," *Artificial Intelligence in Education*, vol. 17, pp. 267–274, 2015.

[41] P. J. Guo, "Where is the second peak? analyzing mooc retention over time," in *Proceedings of the first ACM conference on Learning @ scale conference*. ACM, 2014, pp. 171–180.

[42] R. Rosés, G. Vecchi, B. Markines, and A. Abián, "Using text mining and sentiment analysis for online forums hotspot detection and forecast," *Decision Support Systems*, vol. 89, pp. 1–17, 2016.

[43] S. Maldonado, R. Weber, and J. Basak, "Predicting student failure at school using machine learning," *PloS one*, vol. 14, no. 11, 2019.

[44] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein, "Deep knowledge tracing," in *Advances in Neural Information Processing Systems*, 2015, pp. 505–513.

[45] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[46] Q. Hu and H. Rangwala, "Academic performance estimation with attention-based graph convolutional networks," *arXiv preprint arXiv:2001.00632*, 2019.

[47] H. Li, H. Wei, Y. Wang, Y. Song, and H. Qu, "Peer-inspired student performance prediction in interactive online question pools with graph neural network," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2589–2596.

[48] B. Albreiki, T. Habuza, and N. Zaki, "Extracting topological features to identify at-risk students using machine learning and graph convolutional network models," *International Journal of Educational Technology in Higher Education*, vol. 20, no. 1, pp. 1–22, 2023.

[49] E. Prihar and I. Heffernan, Neil T, "Edm cup 2023," Jun 2023. [Online]. Available: osf.io/yrwuh

[50] W. Mendenhall, R. J. Beaver, and B. M. Beaver, *Introduction to Probability and Statistics*. Cengage Learning, 2012.

[51] E. B. Gregori, J. Zhang, C. Galván-Fernández, and F. de Asís Fernández-Navarro, "Learner support in moocs: Identifying variables linked to completion," *Computers & Education*, vol. 122, pp. 153–168, 2018.

[52] J. Whitehill, J. Williams, G. Lopez, C. Coleman, and J. Reich, "Beyond prediction: First steps toward automatic intervention in mooc student stopout," *Available at SSRN 2611750*, 2015.

[53] T. Daradoumis, R. Bassi, F. Xhafa, and S. Caballé, "A review on massive e-learning (mooc) design, delivery and assessment," in *2013 eighth international conference on P2P, parallel, grid, cloud and internet computing.* IEEE, 2013, pp. 208–213.

[54] A. T. CORBETT and J. R. A. DERSON, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Modeling and User-Adapted Interaction*, vol. 4, pp. 253–278, 1995.

[55] K. VanLehn, "The behavior of tutoring systems," *International journal of artificial intelligence in education*, vol. 16, no. 3, pp. 227–265, 2006.

[56] C. Romero, J. R. Romero, J. M. Luna, and S. Ventura, "Mining rare association rules from e-learning data," in *Educational Data Mining 2010.* ERIC, 2010.

[57] S. Raschka, "Mlxtend: Providing machine learning and data science utilities and extensions to python's scientific computing stack," *The Journal of Open Source Software*, vol. 3, no. 24, Apr. 2018. [Online]. Available: https://joss.theoj.org/papers/10.21105/joss.00638

[58] R. Agrawal, R. Srikant *et al.*, "Fast algorithms for mining association rules," in *Proc. 20th int. conf. very large data bases, VLDB*, vol. 1215. Santiago, Chile, 1994, pp. 487–499.

[59] M. Hegland, "The apriori algorithm–a tutorial," *Mathematics and computation in imaging science and information processing*, pp. 209–262, 2007.

[60] W. L. Hamilton, *Graph representation learning.* Morgan & Claypool Publishers, 2020.

[61] R. Burioni and D. Cassi, "Random walks on graphs: ideas, techniques and results," *Journal of Physics A: Mathematical and General*, vol. 38, no. 8, p. R45, 2005.

[62] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[63] E. K. Ampomah, Z. Qin, and G. Nyame, "Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement," *Information*, vol. 11, no. 6, p. 332, 2020.

[64] S. Cui, A. Sudjianto, A. Zhang, and R. Li, "Enhancing robustness of gradient-boosted decision trees through one-hot encoding and regularization," *arXiv preprint arXiv:2304.13761*, 2023.

[65] J. Gursky, "Boosting showdown: Scikit-learn vs xgboost vs lightgbm vs catboost in sentiment classification," https://towardsdatascience.com/boosting-showdown-scikit-learn-vs-xgboost-vs-lightgbm-vs-catboost-in-sentiment-classification-f7c7f4 2020, accessed: 2023-07-07.

[66] J. Korstanje, "Gradient boosting with xgboost and lightgbm," in *Advanced Forecasting with Python: With State-of-the-Art-Models Including LSTMs, Facebook's Prophet, and Amazon's DeepAR.* Springer, 2021, pp. 193–205.

[67] T.-N. Do, P. Lenca, S. Lallich, and N.-K. Pham, "Classifying very-high-dimensional data with random forests of oblique decision trees," *Advances in knowledge discovery and management*, pp. 39–55, 2010.

[68] C. Zhang, W. Wang, L. Liu, J. Ren, and L. Wang, "Three-branch random forest intrusion detection model," *Mathematics*, vol. 10, no. 23, p. 4460, 2022.

[69] M. Shah, H. Kantawala, K. Gandhi, R. Patel, K. A. Patel, and A. Kothari, "Theoretical evaluation of ensemble machine learning techniques," in *2023 5th International*

*Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE, 2023, pp. 829–837.

[70] 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*, vol. 12, pp. 2825–2830, 2011.

[71] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 785–794. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785

[72] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017.