Journal of Physics:
Conference Series

PAPER • OPEN ACCESS

A curriculum recommendation system using a vector database for challenge-based learning

To cite this article: O B Akhiiezer *et al* 2025 *J. Phys.: Conf. Ser.* **3105** 012023

A curriculum recommendation system using a vector database for challenge-based learning

O B Akhiiezer[1], O A Haluza[1], L M Lyubchyk[1] and V Y Sokol[1,2]

[1] National Technical University "Kharkiv Polytechnic Institute", 2 Kyrpychova Str., Kharkiv, 61002, Ukraine

[2] Department of Computer Science, RWTH Aachen University, Templergraben 55, 52062 Aachen, Germany

E-mail: Olena.Akhiiezer@khpi.edu.ua, Oleksii.Haluza@khpi.edu.ua, Leonid.Liubchyk@khpi.edu.ua, Volodymyr.Sokol@khpi.edu.ua

Abstract. This paper proposes a methodology and describes the implementation of a graph-based curriculum recommender system for flexible, personalized, challenge-based learning. The central element of the proposed recommendation process is an algorithm that uses mixed integer programming to optimize curriculums based on multiple constraints, including logical sequencing and credit accumulation requirements. A procedure for assessing grades based on preliminary clustering of students using the k-nearest neighbors method is proposed. The key to implementing the proposed approach is generating performance vectors and estimating grades for tasks that have not yet been completed by students, which is necessary for personalizing learning recommendations. The architecture of a recommender system based on a vector database, the features of the algorithm for generating recommendations, and the results of assessing the degree of accuracy of estimates are described; generated synthetic data were used to conduct computational experiments.

1. Introduction

Within the continuously evolving domain of higher education, there is an imperative demand for pedagogical innovations that facilitate academic excellence and equip students to navigate the complexities of today's global and professional landscapes. Challenge-based learning (CBL) is an educational approach that engages students in real-world problem-solving, promoting learning and developing critical thinking, creativity, and collaboration skills. Unlike traditional learning models, CBL is student-centered pedagogical conception and focused on real-world problems, requiring students to go through research, solution development, and implementation stages, often working in teams. Through this process, students not only acquire subject matter knowledge, but also develop essential soft skills such as communication, teamwork, and problem-solving. CBL's hands-on approach also increases student motivation and retention, as students see the direct impact of their efforts on solving real-world problems. By preparing students to tackle the challenges of future with confidence and competence, challenge-based learning represents a forward-thinking shift in education that meets the needs of a rapidly changing world [1].

CBL is conceptualized as an innovative pedagogical strategy based on immersing students in real-world, practical problems that require collaborative, creative teamwork and critical thinking to find solutions. Innovation in CBL encompasses both a mercantile aspect, focusing on creating new products, and a social innovation aspect aimed at generating social and cultural values. CBL is becoming one of the key methods for developing competencies and self-learning abilities in students of knowledge-intensive university specialties. Student-oriented learning is growing in the following areas: the formation of new pedagogical approaches, technological integration, and interaction with industry and development support [2–4].

The effectiveness of the implementation and improvement of CBL systems is largely due to the use of modern educational information technologies, among which, in this context, a special place is occupied by recommender systems for supporting academic choice in the learning process. As society increasingly demands specialized skills, education must adapt to each individual's unique interests and abilities. Therefore, user-centered curricula are becoming paramount in today's educational environment to provide a personalized learning experience that promotes growth and mastery. Curriculum recommendation systems are emerging as a valuable tool in the educational landscape, offering personalized guidance to students in their learning journey. These systems, leveraging data-driven techniques, are designed to analyze individual learning progress and preferences, enabling the creation of tailored curriculum and helping students optimize their course selection to achieve their learning goals, providing support for the choice of an individual educational trajectory based on the combination of the interests of the student with the general requirements for the construction of the curriculum, taking into account current and predicted assessments of educational achievement [5].

The introduction of artificial intelligence (AI) into educational recommendation systems represents a transformative approach to personalizing the learning experience. AI algorithms can analyze vast amounts of data on student behavior, preferences, and performance to create highly personalized content recommendations. Using vector databases that excel at processing complex, multidimensional data, these systems can efficiently process and store embeddings generated by AI models containing data on students' current competencies, find learning paths, and provide suggestions to shape each student's unique learning path. This approach not only increases the relevance of recommended resources but also adapts to changing student needs, thereby contributing to a more effective educational experience.

This paper proposes a methodology and describes the implementation of a graph-based curriculum recommender system for personalized CBL. The central element of the proposed recommendation process is an algorithm that uses mixed integer programming (MIP) to optimize curriculums based on multiple constraints, including logical sequencing and credit accumulation requirements. A procedure for assessing grades based on preliminary clustering of students using the k-nearest neighbors (kNN) method is proposed. The key to implementing the proposed approach is generating performance vectors and estimating grades for tasks that have not yet been completed by students, which is necessary for personalizing learning recommendations. The architecture of a recommender system based on a vector database, the features of the algorithm for generating recommendations, and the results of assessing the degree of accuracy of estimates are described; generated synthetic data were used to conduct computational experiments.

2. State of the art

There is extensive literature devoted to the issues of development, implementation and evaluation of the effectiveness of recommender systems in education; a detailed review is given in [5]. Li and Ye [6] presents a personalized online education platform based on a collaborative filtering algorithm, developed using cross-platform compatible HTML5 and high-performance hybrid programming framework approach. A methodology for automated development of individualized curricula is presented in [7], the proposed system combines ideas from graph theory, performance modeling, machine learning, explainable recommendations, and an intuitive user interface. The application of supervised machine learning methods such as decision tree, random forests and support vector machine to predict learning success is discussed in [8]. Furthermore, the study focused on the importance of input features, which had a great impact on simplifying the overall classification process by reducing the number of input features, which consequently led to a better result. To develop methods for improving the efficiency of learning, a hierarchical model of progressive improvement (CHPI) is developed in [9], which is based on the methods of cluster analysis and context-sensitive data envelopment analysis (DEA). CHPI clusters the ontological features of students, uses the context-sensitive DEA method to stratify students of different classes, and calculates measures such as obstacles to determine the control path for individuals with ineffective learning processes. Learning strategies are determined according to the gap between the ineffective individual to be improved and the individuals on the control path. Hlazunova and Ponzel [10] analyzes the main problems in the implementation of a recommendation system for building an individual study plan for a higher education student identifies the main ways to implement such a system, and builds a basic algorithm for recommendation system that can provide top disciplines for a student to study based on his or her previous learning outcomes, the learning outcomes of students of the same specialty for a certain period, and the factor of similarity of subjects and the choice of his or her specialization. Delahoz-Domínguez and Hijón-Neira [11] presents the results of developing a system for providing personalized and contextually relevant recommendations for choosing a university degree. The system aims to optimize the match between student profiles and potential academic programs using advanced machine learning models, including XGBoost, Random Forest, GLMNET, and kNN. Roepke et al. [12] presents the AIStudyBuddy project, with a focus on using rule-based AI and process analytics to support learning planning and cohort monitoring. The concept of a reference architecture and data model for learning path analytics is presented, as well as details on developing custom applications, to ensure quality in learning planning and monitoring of student cohorts.

Of particular interest are works devoted to the use of machine learning methods in the problems of building educational recommendation systems. Abyaa et al. [13] investigates the feasibility of predicting a learner's personality based on educational data through supervised learning algorithms. The study underscores the potential of educational data in enhancing adaptive learning systems by tailoring learning paths and resources according to learners' personality traits, thereby mitigating negative learning outcomes associated with certain personality aspects. The "RSU-ML-PL" algorithm [14] was developed to improve individual learning outcomes in an e-learning environment by identifying at-risk learners and providing personalized self-guided learning programs. Five algorithms, logistic regression, k-nearest neighbors, decision trees, random forest classifier, and support vector machine, are implemented to build a classification model that identifies learners who need additional support. Riad et al. [15] proposes a new adaptive e-learning system that enhances recommendation precision and quality by incorporating learners' motivation levels alongside content-based filtering techniques. An important component of educational recommendation systems is information technologies for assessing and predicting student grades. Badal and Sungkur [16] provide a predictive model to forecast students' performance (grade/engagement) and to analyze the effect of online learning platform's features. The model created in this study made use of machine learning techniques to predict the final grade and engagement level of a learner. Nachouki et al. [17] proposed a model based on random forest methodology to predict students' course performance using seven input predictors and find their relative importance in determining the course grade.

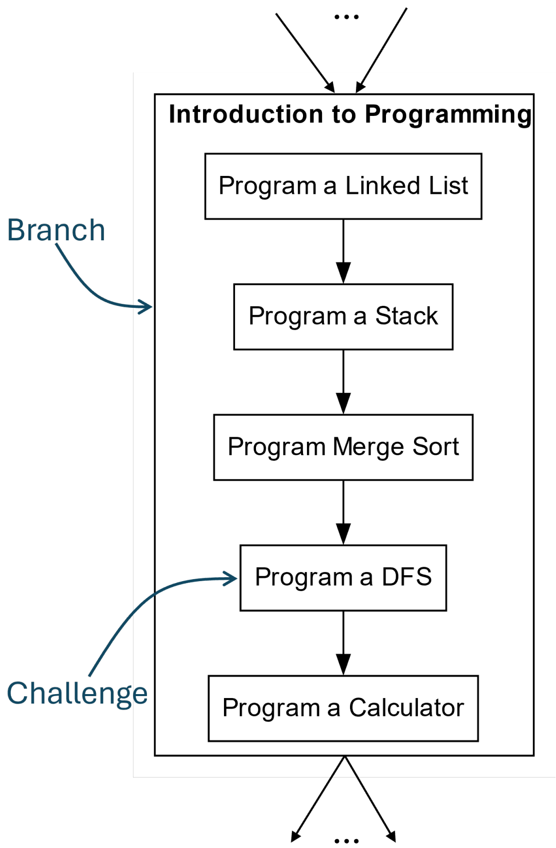
3. Graph course model

The construction of recommendation systems that support students' choice of individual educational trajectories first requires the building of mathematical models of educational programs and courses that describe the composition and sequence of educational tasks and

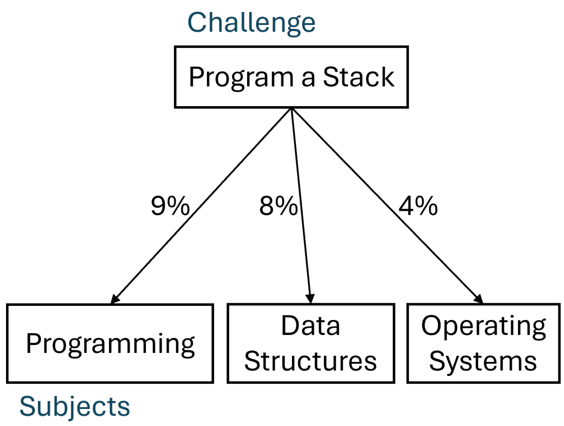
Journal of Physics: Conference Series **3105** (2025) 012023 doi:10.1088/1742-6596/3105/1/012023

work. A course is described by two main components: branches and tasks. In this context, a “branch” is analogous to one thematically dedicated part of a course and includes several “tasks” representing an assignment that must be completed either by an individual student or by a group. The tasks within a branch are arranged in a certain sequence, with the completion of the branch depending on the successful completion of the last task in the prescribed order. An example of a branch with some related tasks is shown in figure 1(a). Individual branches in a course system can be conceptualized as “aggregated” nodes that together form a larger “branch graph”. The key principle in this framework is the precondition requirement: a branch can only be chosen if all its previous branches have been completed. The graph defined by these branches is a directed acyclic graph, describing a one-way progression without cycles, providing a linear path of academic progress for students.

In the context of designing a course system, it is important to establish a mapping between individual tasks and broader academic subjects. To ensure compliance with these rules, a system is needed to track students' progress across the broad subjects. In practice, this involves assigning each task to one or more subjects, together with an assessment of the extent to which completing the task contributes to the requirements of the subject. For example, as illustrated in figure 1(b), the topic “Program the Stack” task can be associated with multiple educational items, each with a different percentage contribution.



(a) Branch and associated challenges.



(b) Subject mapping to challenges.

Figure 1. Examples for challenge-based approach.

Within the graph-based course model, a recommendation is conceptualized as a selection of challenges. The recommendation framework operates under a set of specific constraints to ensure that the recommendations are both practical and effective:

1. **Graph constraints:** Recommendations must adhere to the pre-established graph structure, respecting both the hierarchical order of branches and the sequential arrangement of challenges within each branch.
2. **Target subjects:** The recommendation must encompass a predetermined set of educational subjects that are required for completion, in line with educational objectives.
3. **Total credit count:** There is a minimum credit count that the recommendation should fulfill, ensuring that the student meets academic credit requirements.
4. **Weights for challenges:** Every challenge is assigned a weight, indicative of its anticipated difficulty level for the student.

An objective of the recommendation process is to minimize the total weight of the selected challenges, aiming to balance the academic load effectively.

4. Recommendation system architecture

The recommendation system was designed as a REST API, prioritizing ease of integration with other services. This API serves as the interface for generating personalized recommendations for individual users. We chose Flask (`https://flask.palletsprojects.com/en/3.0.x/`) for implementing the REST API and PostgreSQL (`https://www.postgresql.org`) for the database, based on their robustness and compatibility with our requirements.

Additionally, to enhance user experience and provide a tangible demonstration of the system's capabilities, we developed a simple frontend using React. This not only serves as an interface for the recommendation system but also visually represents the branch graph.

For database interactions, we opted for an Object-Relational Mapping (ORM) approach, using SQL Alchemy (`https://www.sqlalchemy.org`). This decision was driven by the desire for a streamlined and efficient process for handling database operations. A significant aspect of the development was the splitting of the recommendation algorithm into two distinct parts: firstly, the calculation of weights for each challenge, and secondly, the optimization problem solution. One of the most notable was the decision against implementing an explicit caching mechanism. Instead, we leveraged a vector database extension Pgvector (`https://github.com/pgvector/pgvector`) of PostgreSQL to store user embeddings. These embeddings, used to query similar users, significantly reduce database traffic as they do not require re-calculation with every recommendation process. Initially, a caching solution might have been considered for this purpose; however, integrating this data directly into the database presents advantages in terms of querying efficiency and reduces the complexity associated with maintaining a separate cache. Figure 2 offers the recommendation system architecture.

5. Recommendation algorithm

In tackling the task of recommending challenges within the graph model, an algorithmic approach was employed works according to a two-part procedure:

1. **Data preprocessing and grade estimation:** Initially, we estimate grades for all challenges that the user has not yet completed, setting the foundation for personalized challenge recommendations. We focus on the task of assigning an estimated value to each challenge, serving as a metric that predicts a user's performance on that specific challenge.
2. **Recommendation computing via MIP:** The optimal set of tasks is determined using MIP, enabling to identification of the most suitable challenges to recommend to student based on their unique learning path and performance. The objective of MIP is to determine

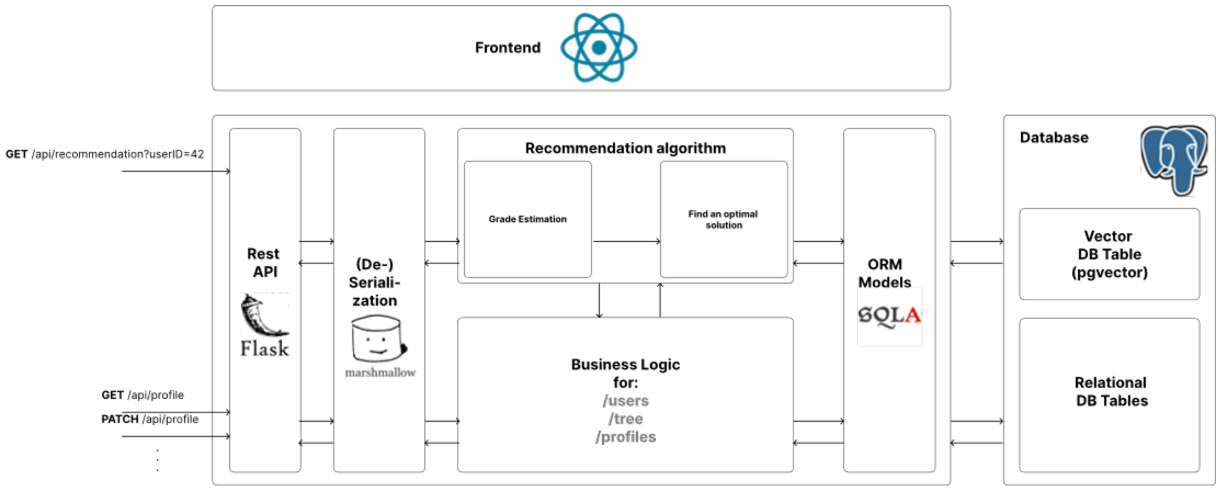


Figure 2. Final architecture of recommendation system.

the optimal subset within the challenge graph while adhering to a defined set of constraints. We defined that optimal selection coincides with an overall optimal average grade on the predicted set of challenges.

The grade evaluation stage of the recommendation algorithm is the part that inputs personalized information from users in the recommendation generation process. The students are pre-clustered using kNN, and the averaged grades of the k-nearest neighbors are then used as the evaluation score for the task. In the final step, we derive a specific estimation for the target by computing the average of the values along the target dimension within the selected k-nearest objects. The historical performance data of students is used as individual records in the vector embedding to generate the grades. For performance improvement, we implemented a technique known as vector imputation. The objective of this method is to populate empty dimensions with meaningful values. We considered four distinct imputation strategies: Mean, Median, kNN, and Iterative Imputation. The mean and median imputation strategy involves filling empty values with the mean or median of the entire dataset within that dimension.

The rating scores for each task are used to assign a difficulty factor to the task. Specifically, a lower rating prediction (indicating a higher level of difficulty) is mapped to a higher difficulty factor in the MIP model, and conversely, a higher rating prediction (indicating ease of execution) is associated with a lower difficulty factor. The goal of the MIP algorithm is to select a set of tasks that, when combined, minimize the total sum of the difficulty factors. The selection process is carried out subject to a set of constraints:

1. **Graph constraints:** For a task selection to be considered valid, it must satisfy the constraints of the graph model structure, namely, the next task can only be selected if all tasks that precede it, either directly or indirectly (i.e., in the previous branch), have also been completed.
2. **Minimum total credits:** The curriculum recommendation must meet the requirement that the total credits associated with the chosen tasks exceed the established threshold.
3. **Target subjects:** Within the algorithm, we have incorporated the concept of configurable target subjects as another key feature. These can be likened to obligatory courses that a student is required to take. This design affords us the versatility to provide user-centric recommendations while introducing a level of rigidity, ensuring essential subjects are included based on specific requirements.

Journal of Physics: Conference Series **3105** (2025) 012023 doi:10.1088/1742-6596/3105/1/012023

In terms of technical aspects, it's noteworthy that we used a solver-independent language for MIP realization. The library facilitating this functionality is *pyomo* [18], offering the advantage of decoupling the solver implementation in the background, allowing for flexibility in selecting different solvers; currently, our implementation utilizes an open-source CBC solver COIN-OR CBC Optimization Suite (`https://zenodo.org/records/13347261`).

6. Vector database

A vector database is designed to efficiently store and retrieve vectorized data. Due to the distinctive data format supported by a vector database, it enables highly efficient querying of vectors, especially in the context of similarity search. Using a vector database to implement a recommendation system provides the following benefits:

1. The need for a separate kNN implementation is eliminated by leveraging the highly efficient internal similarity search capability of the vector database. This allows searching for similar vectors quantified by metrics such as $L2$ distance or cosine similarity. In essence, this built-in functionality serves as an equivalent of kNN.
2. Storing data in a highly aggregated form encapsulated by a performance vector allows implementation of a compact embedding characterizing a student's academic profile. This approach saves network resources and minimizes the number of database queries.

7. Grade prediction assessment

We have implemented a concept aimed at consolidating a student's performance into a unified vector. The performance vector is designed to encapsulate a numeric value for each subject, with a higher value indicative of superior performance. To ensure a standardized representation, we have normalized the performance values, scaling them between zero and one.

A significant hurdle is the scarcity of real-world data. Consequently, there was a pressing need to generate synthetic data that closely mimics real-world scenarios, for testing and refining the recommendation algorithm. To address this, we crafted an example branch graph, the design of the challenges and branches in this graph drew inspiration from various existing study plans, ensuring semantic coherence and relevance. The constructed graph, representing the course structure, includes 200 challenges across 38 branches and covers 39 subjects. The more intricate aspect of our data generation process involved creating student histories. The overarching objective here was to ensure that a student's grades are reflective of a diverse range of students at different stages of their academic journey. This process was divided into two stages:

1. **Generation of expected performance:** For each student, we generate an expected performance level across all subjects. The generation of expected performance for each student hinges on ensuring correlated performance across subjects while allowing for the customization of means and standard deviations for different subjects.
2. **Generation of student histories:** Building on the expected performance levels, we then generated user histories for each student. This stage involved assigning grades for challenges based on the expected subject performance, thereby creating a realistic and varied set of academic records. Two variables are introduced to enhance the realism of the grading process: Challenge Difficulty and Student Performance Variability.

We crafted a tailored framework (figure 3) to assess the quality of grade estimation. The assessment framework comprises three pivotal stages:

1. Performance vector calculation.
2. Search for similar users and estimation of performance.
3. Evaluation of the estimation.

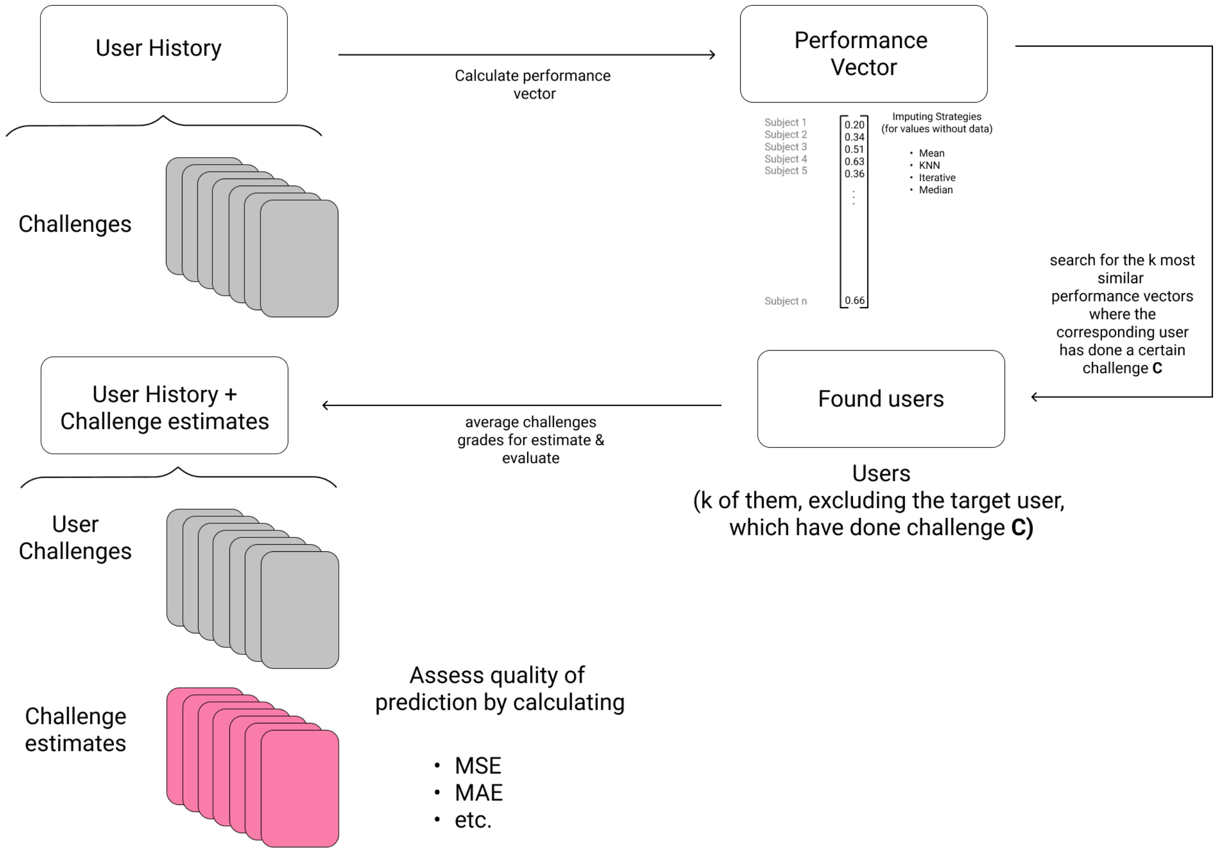


Figure 3. Assessment framework for grade estimation.

At initial stage, we calculate performance vectors for the entire dataset, encompassing all users. This computation is parameterized by the imputation strategies. Using the performance vectors obtained in the initial stage, we proceed to identify a set of users closely situated to the target user. The objective is to determine a balance that avoids underfitting, which occurs when k is too low. Conversely, opting for a higher k may result in overfitting, where the model becomes excessively tailored to users who are inherently distant from the target. We iterate through the process of grade estimation for each challenge previously undertaken by the user, generating outcomes that serve as input for the final stage. To assess the similarity between the real and predicted performance, we have chosen to employ two distinct metrics, namely, MSE and MAE.

Figure 4 depicts the outcomes of the grade estimation utilizing both MSE and MAE.

In the process of computational experiments, the influence of clustering parameters on the accuracy of grade prediction was investigated. The impact of imputation strategies, Mean, Median, kNN, and Iterative Imputation, was also analyzed. Upon examining the results, it is evident that predictions obtained are quite accurate.

8. Conclusions

Considering the outcomes presented in the paper, it is evident that we have successfully achieved our objectives. In future work, we recommend revisiting the grade estimation framework to refine the computation process. While averaging serves as a good starting point, it may fall short of fully incorporating additional knowledge, leaving room for further enhancements. Replacing the mean computation with a weighted kNN approach might already improve the whole system.

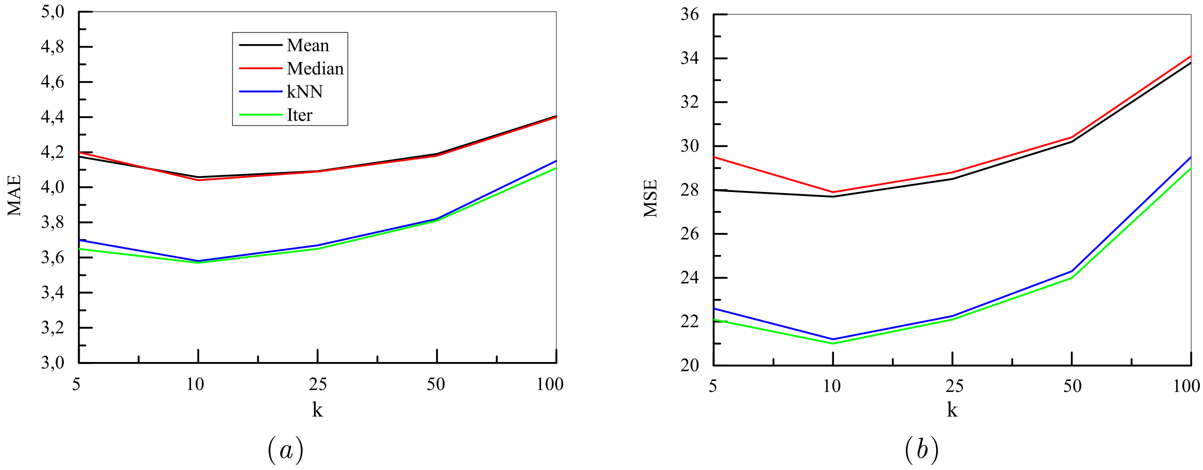


Figure 4. Results of the grade estimation assessment for MAE (a) and MSE (b) for $n = 512$.

We also recognize opportunities for improvements in the non-functional performance of the system. While the grade estimation process, taking approximately 600 ms, is not considered slow for a recommendation algorithm, real-time usage may benefit from additional optimizations. Implementing caching mechanisms or refining the database querying process could potentially enhance the system's responsiveness, making it more efficient for real-time utilization. Moreover, integrating timing constraints could make the system more realistic for practical applications.

ORCID iDs

O B Akhiiezer https://orcid.org/0000-0002-7087-9749
O A Haluza https://orcid.org/0000-0003-3809-149X
L M Lyubchyk https://orcid.org/0000-0003-0237-8915
V Y Sokol https://orcid.org/0000-0002-4689-3356

References

[1] Portuguez Castro M and Gómez Zermeño M G 2020 Challenge Based Learning: Innovative Pedagogy for Sustainability through e-Learning in Higher Education *Sustainability* **12**(10) 4063 ISSN 2071-1050 DOI https://doi.org/10.3390/su12104063

[2] Leijon M, Gudmundsson P, Staaf P and Christersson C 2022 Challenge based learning in higher education–a systematic literature review *Innovations in Education and Teaching International* **59**(5) 609–618 DOI https://doi.org/10.1080/14703297.2021.1892503

[3] Galdames-Calderón M, Stavnskær Pedersen A and Rodriguez-Gomez D 2024 Systematic Review: Revisiting Challenge-Based Learning Teaching Practices in Higher Education *Education Sciences* **14**(9) 1008 ISSN 2227-7102 DOI https://doi.org/10.3390/educsci14091008

[4] Akhiiezer O B, Haluza O A, Savchenko A O, Lyubchyk L M, Protsay N T and Aslandukov M O 2023 Methodology of project-based learning for training junior students in applied mathematics: general scheme of the educational process *Journal of Physics: Conference Series* **2611**(1) 012005 DOI https://doi.org/10.1088/1742-6596/2611/1/012005

[5] Kamal N, Sarker F, Rahman A, Hossain S and Mamun K A 2024 Recommender System in Academic Choices of Higher Education: A Systematic Review *IEEE Access* **12** 35475–35501 DOI https://doi.org/10.1109/ACCESS.2024.3368058

[6] Li J and Ye Z 2020 Course Recommendations in Online Education Based on Collaborative Filtering Recommendation Algorithm *Complexity* **2020**(1) 6619249 DOI https://doi.org/10.1155/2020/6619249

[7] Atalla S, Daradkeh M, Gawanmeh A, Khalil H, Mansoor W, Miniaoui S and Himeur Y 2023 An Intelligent Recommendation System for Automating Academic Advising Based on Curriculum Analysis and Performance Modeling *Mathematics* **11**(5) 1098 ISSN 2227-7390 DOI https://doi.org/10.3390/math11051098

[8] Zayed Y, Salman Y and Hasasneh A 2022 A Recommendation System for Selecting the Appropriate Undergraduate Program at Higher Education Institutions Using Graduate Student Data *Applied Sciences* **12**(24) 12525 ISSN 2076-3417 DOI `https://doi.org/10.3390/app122412525`

[9] Zhao L T, Wang D S, Liang F Y and Chen J 2023 A recommendation system for effective learning strategies: An integrated approach using context-dependent DEA *Expert Systems with Applications* **211** 118535 ISSN 0957-4174 DOI `https://doi.org/10.1016/j.eswa.2022.118535`

[10] Hlazunova O H and Ponzel Y Y 2024 Technologies and algorithms for the implementation of the recommendation system for creating an individual study plan for a higher education student *Proceedings of the 3rd Workshop on Digital Transformation of Education (DigiTransfEd 2024) co-located with 19th International Conference on ICT in Education, Research and Industrial Applications (ICTERI 2024), Lviv, Ukraine, September 23-27, 2024* (*CEUR Workshop Proceedings* vol 3771) ed Vakaliuk T A, Osadchyi V V and Pinchuk O P (CEUR-WS.org) pp 110–117 URL `https://ceur-ws.org/Vol-3771/paper03.pdf`

[11] Delahoz-Domínguez E J and Hijón-Neira R 2024 Recommender System for University Degree Selection: A Socioeconomic and Standardised Test Data Approach *Applied Sciences* **14**(18) 8311 ISSN 2076-3417 DOI `https://doi.org/10.3390/app14188311`

[12] Roepke R, Judel S and Schroeder U 2024 Study path analyses for quality assurance and support of study planning *Informatik Spektrum* **47**(3) 97–104 ISSN 1432-122X DOI `https://doi.org/10.1007/s00287-024-01574-y`

[13] Abyaa A, Idrissi M K and Bennani S 2018 Predicting the learner's personality from educational data using supervised learning *Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications* SITA'18 (New York, NY, USA: Association for Computing Machinery) p 19 ISBN 9781450364621 DOI `https://doi.org/10.1145/3289402.3289519`

[14] Kanokngamwitroj K and Srisa-An C 2022 Personalized Learning Management System using a Machine Learning Technique *TEM Journal* **11**(4) 1626–1633 DOI `https://doi.org/10.18421/TEM114-25`

[15] Riad M, Gouraguine S, Qbadou M and Aoula E S 2023 Towards A New Adaptive E-learning System Based On Learner's Motivation And Machine Learning *2023 3rd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)* pp 1–6 DOI `https://doi.org/10.1109/IRASET57153.2023.10152884`

[16] Badal Y T and Sungkur R K 2023 Predictive modelling and analytics of students' grades using machine learning algorithms *Education and Information Technologies* **28**(3) 3027–3057 ISSN 1573-7608 DOI `https://doi.org/10.1007/s10639-022-11299-8`

[17] Nachouki M, Mohamed E A, Mehdi R and Abou Naaj M 2023 Student course grade prediction using the random forest algorithm: Analysis of predictors' importance *Trends in Neuroscience and Education* **33** 100214 ISSN 2211-9493 DOI `https://doi.org/10.1016/j.tine.2023.100214`

[18] Bynum M L, Hackebeil G A, Hart W E, Laird C D, Nicholson B L, Siirola J D, Watson J P and Woodruff D L 2021 *Pyomo — Optimization Modeling in Python* (*Springer Optimization and Its Applications* vol 67) (Charm: Springer) DOI `https://doi.org/10.1007/978-3-030-68928-5`