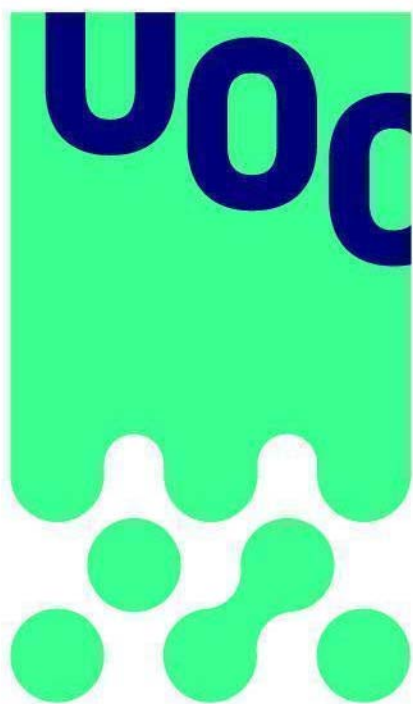


Smart Learning and Assessment System (SLASys): Conceiving an innovative digital training system for Intellectual Property

Final Report

EPO Academic Research Programme 2021

SB-RA2-TA3: Relevance of intelligent tutoring systems to IP education



R&I

research.uoc.edu

Universitat Oberta de Catalunya

Authors:

David Bañeres

Xavier Baró

M. Elena Rodríguez

Ana Elena Guerrero Roldán

Vanessa Jiménez Serranía

Silvia Sivera

Josep Prieto

Universitat Oberta de Catalunya





Table of Contents

Abstract	3
Executive Summary	4
1. Introduction	5
2. State of the art	6
2.1. Large Language Models	6
2.2. BERT Large Language Model	6
2.3. BERT domain-specific application in education	7
2.4. Early Warning System	8
2.5. At-risk failure identification	8
2.6. At-risk dropout identification	9
2.7. Evaluation metrics	10
3. Talent Academy	12
3.1. Preliminaries	12
3.2. Objectives	12
3.3. Guidelines for best practices for Course A	12
3.4. Semantic search engine for learning resources	14
3.5. Short-answer questions assessment recommendation	19
4. Patent Academy	26
4.1. Preliminaries	26
4.2. Objectives	29
4.3. An ETL process to extract and anonymize Moodle data	29
4.4. Detect at-risk students of failing a course	30
4.5. Detect at-risk students of dropping out of a course	35
4.6. Next activity recommendation	40
4.6. Course enrollment recommendation	41
5. Slasys system	44
5.1. Design a container-based architecture	44
5.2. Talent Academy	45
5.3. Patent Academy	54
6. Conclusions	60
Bibliography	61

Abstract

This project proposes to use the Design-Science Research methodology to specify, design and develop the Smart Learning Automated System (SLASys) suited explicitly for Intellectual Property (IP) training.

This project aims at:

- Defining a smart learning system to help students succeed in their learning process using artificial intelligence techniques for IP training.
- Defining the best assessment strategies and learning resources for an automatic feedback system.
- Building different Artificial Intelligence (AI) components to enhance personalization in IP learning.

The project has analyzed the different needs of Talent and Patent Academies. For Talent Academy, the project has been centred on the fundamental Course A. Course A is part of the courses oriented to internal formation for employees and the formation for students for patent examiners. This course has been selected because it has been subjected to substantial amendments in recent years: new learning resources have been drafted, a new learning methodology has been applied, and new activities are offered.

The aim of the project for Course A can be summarized in the next objectives:

- OT1. To create a recommendation report about techniques for effective teaching, learning, assessment, and engagement techniques.
- OT2. To design an integrated framework to track exercises on Course A.
- OT3. To extract the main concepts/skills/keywords of RISE learning resources.
- OT4. To design a search engine for learning resources.
- OT5. To design a learning resources recommender based on theoretical questions.

Patten Academy offers standalone e-learning courses related to IP training. These courses are provided via the Moodle learning management system (LMS), which gathers data about students that can be used to better analyze students' behaviour to support them during the learning process.

The aim of the project for Patent Academy is:

- OP1. To analyze the available Moodle data.
- OP2. To specify and design an Extract-Transform-Load (ETL) process to extract data from the Moodle database.
- OP3. To specify and design predictive models for predicting students' failure.
- OP4. To specify and design a predictive model for recommending the next course to enrol in.
- OP5. To specify and design dashboards for teachers to visualize such data.
- OP6. To specify and design a system to inform different stakeholders (i.e., teachers and students) of the different outputs of the models.

This document aims to describe the outputs of the ARP project.

Executive Summary

This section briefly summarizes the outputs obtained from the project. In the experimental part for Talent Academy, we project achieved the next outputs:

- Recommendation report about best practices for flipped-classroom methodology (Contributes to OT1).
- Design a custom web crawler system for proprietary RISE SCORM packages (C. to OT3).
- Analyze the best-suited approaches to design a semantic search engine for learning resources (C. to OT4).
- Analyze the best-suited approaches to find theoretical question answers on learning resources (C. to OT5).
- Design a semantic search engine for learning resources (C. to OT4).
- Analyze best-suited approaches for assessing short-answer questions (C. to OT2).
- Design a recommender system for assessing short-answer questions (C. to OT2).
- Perform four pilots on real students to test the short-answer correctness recommendation system (C. to OT2).

Additionally, the project produced the following AI artefacts:

- Design a container-based architecture for the SLASys system (C. to all objectives).
- Provide an API to incorporate the search engine on custom systems (C. to OT4).
- Provide two integration examples (custom web-based and Moodle) (C. to OT4).
- Provide an API to assess short-answer questions (C. to OT2).
- Provide two integration examples (custom web-based and Moodle) (C. to OT2).

Related to Patent Academy, the outputs in the experimental part were:

- An ETL process to extract and anonymize Moodle data (C. to OP2 and OP1).
- Data formatting for creating the predictive models (C. to OP2).
- Analyze predictive models for the different course modalities (C. to OP3 and OP4).

Finally, the project produced the following AI artefacts:

- Design the web interface to visualize information for teachers and administrators (C. to OP5).
- Design a recommendation system based on predictive models to support students better (C. OP6).

This document provides more detailed information about the previously described outputs.

1. Introduction

Nowadays, educational institutions (including traditional and online institutions) generally tend to adopt information and communication technologies (ICT) to support their students' learning processes. Many technological systems help students to learn. Some of them aid students in the learning phase by helping them find learning resources or recommend exercises. Others aim to help the student in the assessment phase to give feedback. Furthermore, some others monitor the student's progress during the instructional process to recommend the best learning path to succeed in the course.

In the past, such systems were denoted as Intelligent Tutoring Systems (ITS), which commonly proposed some exercises or activities for training and others for automatically assessing the student. The system's sophistication was in the feedback automation to help students fix mistakes and acquire skills.

However, artificial intelligence (AI) resurgence due to the accessibility to computational resources and specific tools and packages for using AI techniques have evolved classic ITS to enhanced Smart Learning Systems (SLS). New systems are powered by AI algorithms that can train complex predictive models to simulate students' behaviour and anticipate undesirable outcomes. Also, recommendation algorithms can be developed to give instant feedback to students. Furthermore, Natural Language Processing (NLP) advances can even simulate teacher interaction.

However, such systems cannot be straightforwardly used in any discipline or domain. First, the instructional design must be adapted by analysing how students learn and are assessed in that discipline. Second, learning resources should be digitally available and searchable for computerised processes. With these conditions, an SLS will be capable of autonomously running a learning process by assessing students, providing feedback regarding recommendations of learning resources, answering students' questions, and predicting students' performance and acquired knowledge.

Identifying which resources and activities to use in each course will depend on the expected learning outcomes, target audience, and discipline. A combination of learning resources of different natures, together with the use of gamification and virtual reality strategies, may be deployed in several layers of knowledge. Thus, the designed course may be enjoyed (and equally profited) by participants with different backgrounds and expertise. This project intends to provide the basic elements to guide towards a successful combination of different learning resources (text, video, and audio) and activities, as well as an example of an IP course that integrates and applies all these elements. This example will serve as a starting point for building a smart learning system.

We propose to use the Design-Science Research methodology to specify the Smart Learning Automated System (SLASys) suited explicitly for Intellectual Property (IP) training.

This project aims at:

1. Defining a smart learning system to help students succeed in their learning process using artificial intelligence techniques for IP training.
2. Defining the best assessment strategies and learning resources for an automatic feedback system.
3. Building different Artificial Intelligence (AI) components to enhance personalization in IP learning.

2. State of the art

2.1. Large Language Models

Large Language Models (LLMs) have emerged as a transformative technology in recent years, demonstrating remarkable capabilities across a wide range of natural language processing tasks. These models are trained on vast amounts of textual data, allowing them to capture complex linguistic patterns and generate human-like text. The rapid progress in LLMs has led to significant advancements in machine translation, text summarization, question answering, and even code generation.

One of the key factors driving the success of LLMs is the development of advanced neural network architectures, particularly the Transformer model. Transformers (Vaswani et al., 2017) have become the backbone of most state-of-the-art LLMs by using self-attention mechanisms to capture long-range dependencies in the input text, enabling more effective modelling of context and semantics. Recent architectural innovations have further improved the performance of LLMs. For example, the introduction of the GPT (Generative Pre-trained Transformer) family of models, such as GPT-3 (E. Hu et al., 2022) and GPT-4 (OpenAI et al., 2023), has demonstrated impressive learning capabilities, where the model can adapt to new tasks with minimal fine-tuning or task-specific examples.

The training of LLMs is a computationally intensive process that requires vast amounts of computational resources and curated datasets from various sources, including web pages, books, and social media, to capture a wide range of linguistic styles and domains (Candel et al., 2023). This combination has improved the model's ability to perform complex reasoning tasks and its performance and generalization capabilities.

Despite their impressive performance, LLMs also face several limitations and challenges. Due to the generalization capability, LLMs might misinterpret tasks in specific domain languages that require fine-tuning with new data. Another challenge is the interpretability and explainability of LLMs. Due to their complex architectures and vast number of parameters, it can be difficult to understand how these models arrive at their predictions. Finally, LLMs are complex models that require a vast amount of computational resources to work. Thus, researchers continuously explore how such models can be pruned to be functional as private instances. Smaller models have been produced in recent years with high-quality results that can be used on specific-domain tasks (Brown et al., 2020; Raffel et al., 2020).

2.2. BERT Large Language Model

BERT (Bidirectional Encoder Representations from Transformers) is a state-of-the-art NLP model developed by Google in 2018 (Devlin et al., 2019). BERT is a transformer-based language model pre-trained on a large corpus of text data and can be fine-tuned for various NLP tasks. BERT has had a transformative impact on natural language processing, and its continued development and application in various domains demonstrates its versatility and potential for further advancements in AI and language understanding. Additionally, BERT has been improved in recent years by reducing the computational requirements and keeping most functionalities, such as DistilBERT (Sanh et al., 2019) or ALBERT (Lan et al., 2020).

In architectural aspects, BERT is (1) bidirectional to capture contextual information from both the left and right sides of a word, (2) allows transfer learning by fine-tuning the model on specific tasks and datasets, and (3) has a transformer architecture which relies on attention mechanisms instead of recurrent or convolutional neural networks.

BERT has been used widely for a diverse range of NLP tasks, such as text classification (Adhikari et al., 2019), text entity recognition (Souza et al., 2019), natural language inference (A. Wang et al., 2018), language generation (H. Zhang et al., 2019), and domain-specific applications, such as BioBERT (Lee et al., 2020) and SciBERT (Beltagy et al., 2019) for biomedical and scientific text, respectively.

2.3. BERT domain-specific application in education

BERT has also impacted the educational domain. Due to its semantic interpretability, it has a wide range of applications.

The most relevant application is the development of intelligent tutoring systems that understand the students' queries, provide detailed explanations, and adapt learning resources to individual needs. Additionally, learning platforms can be adapted based on students' performance and preferences (Gligorea et al., 2023), recommending specific resources and exercises for weakly acquired knowledge areas. These recommendations can be context-aware or incorporate conversational agents for better personalization and interaction (Kusal et al., 2022). BERT has also been proven to be a good method for summarizing long texts and paraphrasing content (X. Zhang & Lapata, 2017), making it easier for previous functionalities to provide answers, for students to grasp key concepts and for teachers to prepare teaching resources.

LLM has also impacted assessment tasks. Teachers can use BERT to generate questions from existing texts, which can be used for quizzes and exams (Nguyen et al., 2022) and grade them. BERT models can assess student essays' coherence, relevance, and grammar (Y. Wang et al., 2022) by providing scores and detailed feedback and helping students improve their writing skills. Furthermore, short-question answers can also be graded (Burrows et al., 2015), reducing the workload for teachers and ensuring consistency during the assessment process. The students' sentiment and engagement levels can even be analyzed based on feedback and discussions, enabling teachers to make proactive interventions (Truong et al., 2020).

BERT-based applications for language learning were also designed to help students understand idiomatic expressions and cultural nuances (Bahdanau et al., 2015). Specific recommenders or conversational agents are being developed. Chatbots can engage students in natural language conversations, helping them practice speaking and comprehending new languages (Z. Zhang et al., 2019).

Applying LLM solutions in educational contexts faces the same challenges and issues as the general domain. Scalability and integration with existing infrastructures require significant investment and technical expertise (Xu & Zhu, 2023). However, LLMs in education are still evolving, with ongoing research and development to enhance their capabilities and address current limitations. Its potential to transform personalized learning, automate administrative tasks, and support educational research makes it a valuable tool in the modern educational landscape. As the technology matures, it will likely become an integral part of educational systems worldwide, offering more tailored and effective learning experiences.

2.4. Early Warning System

Nowadays, teachers are pushed to provide a better personalized educational experience based on students' needs. Some examples of such personalization are feedback on assessable activities, resource recommendations, and self-regulating guidelines (El-Sabagh, 2021), where LLMs can help. However, these strategies are more effective when they are provided based on students' individual needs, such as when at-risk situations may appear. Although there are many possible at-risk situations, we can summarize them into two possible ones: at risk of failing or dropping out of the enrolled course (Simpson et al., 1980). AI techniques, including statistical models, educational data mining, and machine learning, have a major potential for such at-risk identification (Zawacki-Richter et al., 2019).

Personalization can be achieved by knowing the students better. Learning analytics and educational data mining research areas explore how collected and analyzed data from digital systems can enhance the teaching and learning process (Siemens & Baker, 2012). Both research areas focus on tracking students in the LMS rather than their assessment process or engagement level. Thus, few insights exist about reducing dropout or increasing students' motivation. However, several adaptive systems have been developed to support teachers and students across their courses in recent years. Some are just collecting navigational data when students interact with the LMS in communication spaces like virtual classrooms, debates, or forums, but few focus on students' actions when performing learning activities and assessments (Mousavinasab et al., 2021).

Such an adaptive system is denoted as an Early Warning System (EWS), which tracks students' actions, provides feedback about students' status to prevent dropout, and detects at-risk students to impact students' performance, retention, and satisfaction. There are different kinds of EWS depending on the focus: retention in face-to-face environments (Knowles, 2014; Márquez-Vera et al., 2016), retention in online courses (Lykourantzou et al., 2009; Srilekshmi et al., 2017; Xing et al., 2016), or at-risk early failure detection (Casey & Azcona, 2017; Macfadyen & Dawson, 2010; Vandamme et al., 2007; You, 2016). As stated in different works (Freitas & Salgado, 2020; Ortigosa et al., 2019), many EWS approaches focus on defining predictive models from the available datasets to identify such at-risk conditions (Cerezo et al., 2016; Huang & Fang, 2013; López-Zambrano et al., 2020), but few full-fledged developments can be found. However, the number of developments applied in real educational settings has increased in recent years. Some systems only focus on showing dashboards for teachers (Najdi & Er-Raha, 2016; Wolff et al., 2014). Other systems additionally provide information to students (Y. H. Hu et al., 2014; Ortigosa et al., 2019) since it is essential to inform and empower each stakeholder group.

2.5. At-risk failure identification

At-risk failure is defined as the likelihood of failing a course. Different systematic reviews (Rastrollo-Guerrero et al., 2020; Zawacki-Richter et al., 2019) have explored the different models that can be applied for failure identification. Independently of the desired outcome, models have used many different types of data in order to perform the predictions. Different variables (or features) have been explored, ranging from demographic data (Saarela & Ark Ainen, 2015) (e.g., age, gender, ethnic origin, marital status, among others), self-reported questionnaires (Mishra et al., 2014; Vandamme et al., 2007), continuous assessment results (Arnold & Pistilli, 2012; You, 2016), user-generated content (Saura et al., 2019) to LMS data (Romero et al., 2013; Zacharis, 2015).

Numerous classification algorithms have been analyzed through the proposed predictive models. Decision Tree (Azcona & Casey, 2015), Naive Bayes (Marbouti et al., 2016), Support Vector Machine (Gašević et al., 2016), Logistic Regression (Casey & Azcona, 2017; Macfadyen & Dawson, 2010), Hierarchical Mixed models (Arnold & Pistilli, 2012; Joksimović et al., 2015), K-Nearest Neighbors (KNN) (Casey & Azcona, 2017), Neural Network models (Calvo-Flores et al., 2006), or Bayesian Additive Regressive Trees (Howard et al., 2018) are some examples of the employed techniques.

2.6. At-risk dropout identification

Dropout is a challenging problem in Higher Education (HE), including face-to-face, blended and online settings (Grau-Valldosera & Minguillón, 2014; Stone & O'Shea, 2019; Yair et al., 2020), since it seems an unsolvable problem. Many researchers have analyzed the problem (Greenland & Moore, 2022; Xavier & Meneses, 2020) and the factors (Bağrıacık Yılmaz & Karataş, 2022; Greenland & Moore, 2022; Thalhammer et al., 2022; Tinto, 1975; Xavier & Meneses, 2022) to enlighten the reasons behind the problem, but students are still dropping out.

The first problem is that there is no agreement on the dropout definition (Xavier & Meneses, 2020), and it is sometimes misinterpreted with related concepts (e.g., completion, retention, success, persistence, among others). Moreover, dropout has a temporal conception ranging from a long-term perspective (leaving an academic program or university) to a mid-term perspective (course dropout).

Focusing on course dropout, dropout identification in online settings has been extensively studied in Massive Open Online Course (MOOC) settings (Dalipi et al., 2018; Goel & Goyal, 2020; Moreno-Marcos et al., 2019) due to low retention. MOOC courses use a self-paced learning approach where students can progress through the material at their own speed rather than adhere to a fixed schedule. In such courses, models effectively identify dropout students (Mubarak et al., 2020; Whitehill et al., 2017) since models are trained with unbalanced data where dropout distribution ranges between 70% and 90% on average. Some of these models have been used to develop analytical tools to help teachers to understand when dropout appears and which factors related to engagement and performance impact its materialization (Boudjehem & Lafifi, 2021; Chen et al., 2017; Dourado et al., 2021; Itani et al., 2018; Tang et al., 2015).

Developed models focus on a fixed (and predefined) temporal prediction denoted as FTPred. This approach specifies a fixed temporal interval, e.g., week, activity deadline, or percentage from semester completion (Mubarak et al., 2021; Whitehill et al., 2017) because of the difficulty of identifying dropout students by a daily prediction. These models may have a high accuracy variability, causing significant false positives (i.e., predicting as dropout non-at-risk students).

Several works proposed different FTPred intervals ranging from the activity duration (Alamri et al., 2021; Kotsiantis et al., 2003; Lykourantzou et al., 2009) to weekly predictions (Fei & Yeung, 2016; Mubarak et al., 2020, 2021; Whitehill et al., 2017). Such fixed and predefined periods simplify the dataset construction for training and predicting. In the former, the FTPred size is adjusted to the activity duration, reducing the complexity of data aggregation, and the prediction is issued after the activity is graded. However, the FTPred size can be too large and too late to be considered within an intervention mechanism. In the latter, the data are aggregated by weekly periods. However, most approaches are oriented to MOOC-based courses where the activity duration is usually constrained to one week.

Realtime daily predictions are preferable, but issues related to false positives may still occur. To amend this drawback, confidence interval periods of variable size denoted as dropout temporal window (DTWin) for improving the detection of at-risk students are being proposed (Bañeres et al., 2023). This technique seeks the set of minimum consecutive days a student is predicted as a dropout to consider him/her a real at-risk dropout student. A DTWin approach is adapted for each course and/or activity depending on the duration and complexity of the course and/or activities.

2.7. Evaluation metrics

Models accuracy is commonly evaluated with the same metrics. The different metrics used in this project are described next.

Binary classification models can be evaluated by checking whether the prediction corresponds to the expected value in the testing or validation set. Four accuracy metrics are defined:

$$TNR = \frac{TN}{TN + FP} \quad ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

$$TPR (recall) = \frac{TP}{TP + FN} \quad F_{1.5} = \frac{(1 + 1,5^2)TP}{(1 + 1,5^2)TP + 1,5^2FN + FP}$$

where TP denotes the number of instances of the positive class (i.e., 1) correctly identified, TN the number of instances of the negative class (i.e., 0) correctly identified, FN the number of instances of the positive class incorrectly identified, and FP the number of instances of the negative class incorrectly identified. These four metrics are used for evaluating the global accuracy of the model (ACC), the accuracy when detecting the positive class (true positive rate - TPR), the accuracy when distinguishing the negative class (true negative rate - TNR) and a harmonic mean of the true positive value (precision) and the TPR (recall) that weights correct identification (F score - $F1.5$).

When a model predicts a natural number (i.e., finding the number of days until the student will stop accessing the course), previous metrics are not appropriate. The error between the predicted and correct number of days can be evaluated based on two additional metrics:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - yp_i| \quad RMAE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - yp_i)^2}$$

where n is the number of tested instances, y_i is the correct value, and yp_i is the predicted value. These two metrics define the mean absolute error (MAE) and the root mean square error ($RMSE$). The major difference is that $RMSE$ is more affected by the error between the predicted and the correct value.

For recommender models (i.e., search engine or activity recommendation), another metric is used since the model predicts a set of k values, and the objective is to check whether the correct value is within the list. Two metrics are used:



$$\begin{aligned} & \text{Prec@}k \\ &= \frac{1}{n} \sum_{i=1}^n \frac{\text{Pred. correctly on } k \text{ recom}_i}{k_i} \end{aligned}$$

$$\begin{aligned} & \text{Recall@}k \\ &= \frac{1}{n} \sum_{i=1}^n \frac{\text{Pred. correctly on } k \text{ recom}_i}{\text{Total Relevant}_i} \end{aligned}$$

where n is the number of tested instances, and k_i is the number of top k recommendations for instance i . These metrics analyze the capacity to recommend the correct value, and detect all correct values when there are multiple correct values (i.e., $\text{Total Relevant}_i > 1$), respectively.

AUC metric is also used for enrollment recommendation and evaluates the probability that the recommender ranks a randomly chosen relevant course higher than a randomly chosen non-relevant one.

3. Talent Academy

3.1. Preliminaries

Course A is part of the courses oriented to internal formation for employees and the formation for students for patent examiners. The complete list of courses is based on courses A to G, with a 2-year duration.

Course A is the longest, with a duration of 6 weeks. This course has been selected because it has been subjected to substantial amendments during the past years: new learning resources have been drafted, a new learning methodology has been applied, and new activities are offered. The changes in other courses are ongoing. The objective is to change them in the future.

The objective of the course is to prepare the student for the patent examination. Learning methodology is based on a flipped classroom strategy combined with Just-in-Time teaching. Self-learning activities and reading learning resources are proposed to be performed individually offline. Then, there are also online synchronous sessions with all the students and the teacher, where offline activities are discussed, and practical exercises, questions, and group activities are performed.

The learning resources are in HTML and developed with RISE product (<https://articulate.com/360/rise>), which can be exported as a single HTML page or as a SCORM package. These learning resources are combined with multimedia material for complex concepts. Self-learning activities are offered in Word documents with a large set of practical exercises and short-answer questions.

Currently, there is no supporting LMS. Different communication tools are combined to offer the course. Since there is no LMS, there is no tracking of the knowledge acquired by the students. Additionally, there is no continuous assessment during the course, making assessing the acquired knowledge difficult. The feedback is limited to the discussions during the online synchronous sessions and the students' doubts via communication tools (e.g., email).

In this project, we proposed recommendations to improve Course A's learning process using AI-powered technological tools.

3.2. Objectives

The aim of the project for Course A can be summarized in the next objectives:

- OT1. To create a recommendation report about techniques for effective teaching, learning, assessment, and engagement techniques.
- OT2. To design an integrated framework to track exercises on Course A.
- OT3. To extract the main concepts/skills/keywords of RISE learning resources.
- OT4. To design a search engine for learning resources.
- OT5. To design a learning resources recommender based on theoretical questions.

3.3. Guidelines for best practices for Course A

Aforementioned, Course A in Talent Academy has significantly changed over the last few years. The learning resources have been enhanced from simple PDF to rich web-based format. Additionally, the learning process has changed from face-to-face learning to a flipped classroom methodology, mainly due to the pandemic.

This section focuses on the objective *OT1. To create a recommendation report about techniques for effective teaching, learning, assessment, and engagement techniques*. The learning process has been analyzed jointly with course designers from the Universitat Oberta de Catalunya (UOC), who have a large online course design experience with different learning style modalities. *Deliverable 2* is a guideline for best practices with theoretical recommendations and specific analysis for Course A. This section summarizes the key factors.

The key factor to ensure teacher and student satisfaction with an educational model and methodology must be coherent among all the elements of the chosen model. In virtual modalities, none of these elements can be improvised, as the quality of the training process depends significantly on the course design.

- The technological-pedagogical design of the training process is the axis around which the rest of the dimensions pivot. All of them should be defined in the design stage to work in harmony within the whole model. In the flipped classroom modality, aligning the design of all remaining dimensions is essential for a coherent teaching-learning process.
- Regarding teaching time, it is important to decide what will happen in every phase and to identify which actions will be more meaningful for the student learning process and if they are carried out synchronously or asynchronously. The course designer should consider whether any activities and/or actions currently conducted asynchronously could be carried out synchronously or vice versa, always maintaining a balance between both while assessing these teaching times from a pedagogical perspective.
- The entire technological ecosystem must be integrated into a single virtual environment where students and teachers can find everything they need throughout the training process, delivered according to the technological-pedagogical design. Not having a single virtual environment containing all the elements needed for the teaching-learning process results in inefficient time management for teachers and students, and it does not allow classroom activity to be controlled, recorded, and monitored. This means that certain circumstances that may impact on many levels are not being considered, including class delivery and issues of a legal nature such as data protection, etc.
- Concerning ratios, the number should be according to the effort the teacher can assume. When planning the activities, the training must be designed with the ratio in mind, considering how feedback will be given on each activity, whether this will be given in a more personalized way or in groups to facilitate the task.
- The teacher is a key figure who should take a proactive role in performing their functions. They must have the digital competencies to be able to design the course with integrated technology and be able to transfer this design to the organization of the virtual classroom. Moreover, teachers should take full advantage of technology for student engagement, guidance, and assessment tasks.
- In flipped classrooms, students are asked to be rather proactive and self-sufficient in their learning. This means that the course design needs to be oriented toward aiding their autonomy and time management while fostering meaningful learning. Therefore, from the beginning of the course, students should have access to a single virtual environment that contains all the information about each stage of their training experience, the complete course plan with a calendar of activities, details about the type of activity, the resources that will be available, what assessment criteria will be used, among others.

- Focusing on the content of the learning resources, they should be integrated into the virtual classroom where all the actions relating to the course will be conducted, thus forming the digital ecosystem of the classroom. Additionally, the teachers should be trained to incorporate the educational technology necessary to improve student engagement in both synchronous and asynchronous course sessions.
- Regarding assessment, it is important to have short assessment periods instead of a unique assessment at the end. For students, it is crucial to know whether they are making adequate progress. Personalized feedback is indispensable to any training action to ensure meaningful student learning. Posting the correct solution to an activity without providing feedback on the student's answers can cause confusion in cases where the student does not understand why their answer is incorrect.

3.4. Semantic search engine for learning resources

This section focuses on diverse objectives: OT3. *To extract the main concepts/skills/keywords of RISE learning resources*, OT4. *To design a search engine for learning resources*, and OT5. *To design a learning resources recommender based on theoretical questions*.

Course A learning resources in RISE format are different HTML pages that can be accessed from an HTML index of contents without any search engine. Students have difficulty finding concepts since they must access the different HTML pages until the concept is found. Thus, a search engine is needed to perform queries in the complete set of learning resources. Additionally, the search engine should be capable of answering questions related to the resources.

Different strategies have been explored to extract key concepts from the resources, obtain the answers to questions, and search efficiently. After the different experiments, the search engine was developed with Elastic Search, which allows semantic and literal search (i.e., based on keywords). The project also provides an automated method to index the learning resources for use within the Elastic Search.

This search engine was designed as an API (see Section 5.2.1), allowing integration on different services. This project provided two integration examples: on a web index page and within Moodle LMS by searching for different SCORM packages. Deliverables 1 and 3 summarized this part of the research.

3.4.1. Design a custom web crawler

In order to create the search engine, the contents need to be readable by the system. Therefore, a web crawler was created to extract the information from HTML pages. The crawler accepts an HTML page or a list of pages and extracts the contents. The crawler only extracts information within the server domain where the HTML page is stored. Links to other resources or web pages outside the domain are omitted.

As mentioned, learning resources in Course A are developed with RISE proprietary format (<https://articulate.com/360/rise>). Once the resource is completely edited, the learning resource can be packaged as a unique HTML page or a SCORM package. Technically, the RISE packages have been created with React technology to support graphically enhanced

effects. However, such technology complicates straightforward data extraction since the HTML is encoded in base64 within Javascript.

The web crawler focuses on this type of HTML page and extracts the following information:

- Title of the web page.
- URL path where the web page is stored.
- Subtitle of the subsection: The RISE learning resources are structured in subsections.
- IdSubtitle: The subsection within the material can be directly accessed by adding it to the URL path (i.e., <path>/#/lessons/<idsubtitle> redirects to the subsection).
- Text: Raw text of the web page.

The extracted data within the RISE package is shown in Figure 1.

Extracted data

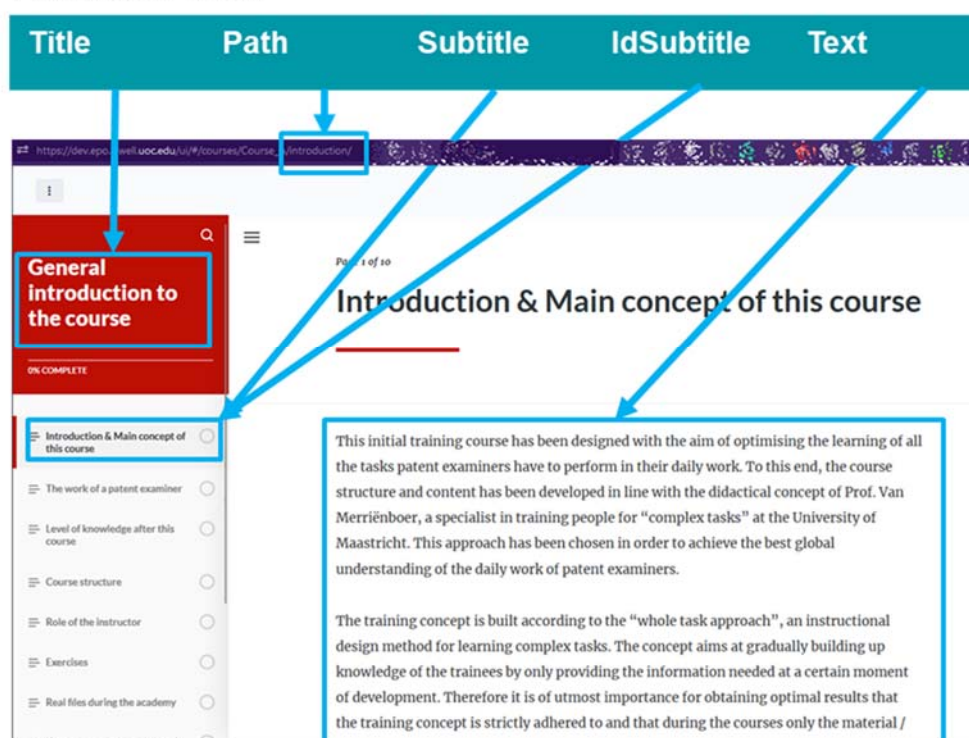


Figure 1. Extracted data within the RISE package.

3.4.2. Analyze best-suited approaches for semantic search

The semantic search consists of finding documents with similar meanings to a provided query text. In contrast, a traditional literal search looks for keywords to decide when the document matches the query. In semantic search, the query and the sentences of the documents are projected in the space of semantic information (i.e., encoded as embeddings that are vectors of numbers). The property of embeddings is that sentences with similar meanings have a close distance between them (i.e., similar numbers). Such property allows quickly finding sentences in documents that have a similar meaning to a search query or even a search question.

The semantic encoding is performed using an LLM. Training such models from scratch requires a huge amount of data and computational resources, which are unavailable for the project. Instead, LLM pre-trained models are fine-tuned using a transfer learning approach with Course A resource documents. This process minimizes the need for specific data, training time, and resources.

Experimental results

The base models were selected by considering the resources required for inference and avoiding needing expensive GPU devices to be deployed in production. We have compared medium-sized bi-directional models based on Google's BERT (Devlin et al., 2019) and related models, such as ALBERT (a lite BERT) (Lan et al., 2020), T5 (Raffel et al., 2020), and ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) (Clark et al., 2020). We have also experimented with Facebook's BART (Lewis et al., 2020), which combines the capabilities of bi-directional and auto-regressive models.

Experimental tests showed similar results between the different models. We fixed the use of BERT as the reference model for experimentation since we found that big issues were not in document representation but in the availability of the required documents.

In addition to the model itself, it is important to consider the distance metric used during the model training. There are mainly two big scenarios:

- The cosine similarity is used when the query text and the target document have similar lengths, when the search is only performed in the document title, and when the search is performed on large documents sentence by sentence.
- The dot product provides better similarity results when both lengths are considerably different. We will consider this scenario when the query text is compared with the complete document content.

We use a question-and-answer (Q&A) dataset extracted from theoretical questions from learning resources from Course A, containing a list of questions, the corresponding answer, and information on the part of the course learning resources related to the question. We compare the effect of the different parameters on the results, considering both literal search based on keywords and semantic search based on BERT encoding. The experimental setup is summarized in Figure 2.

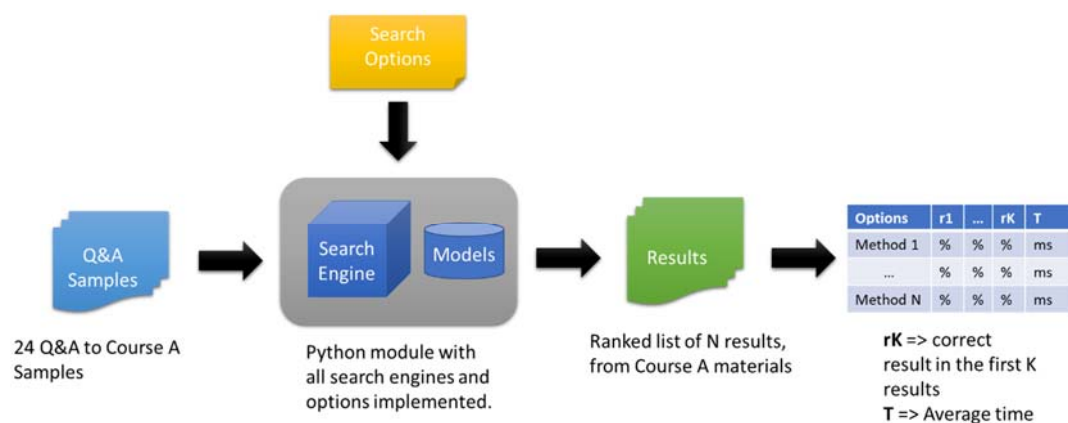


Figure 2. Experimental design representation.

First, we have indexed all the documents for Course A using BERT encodings, including parsed textual content and titles extracted with the previously described web crawler (See Section 3.4.1). Maximal Marginal Relevance (MMR) (Carbonell & Goldstein, 1998) is used to extract relevant keywords from the document. MMR is an information retrieval and text summarization technique that balances relevance and diversity in selecting documents or sentences. Although it is commonly used in sentences, it can also be used to identify the best and most diverse set of keywords that describe a document. MMR enables literal search on keywords for the document content and title.

For each Q&A sample, we used the question (or the answer, depending on the input parameters) to search in the learning resources. The output is a list of sorted results that can be empty. We consider the search failed if it is empty or the correct document is not in the results list. Otherwise, we store the position of the correct document and consider this as the rank of the response.

We compare the use of literal and semantic search, and in both cases, using the whole document or the independent sentences. Figure 3 shows the top 5 combinations encoded as `<semantic/literal>_<document/sentence>_<knn/and/or>_<question/answer>`. The $Prec@k$ metric is used to check accuracy detection.

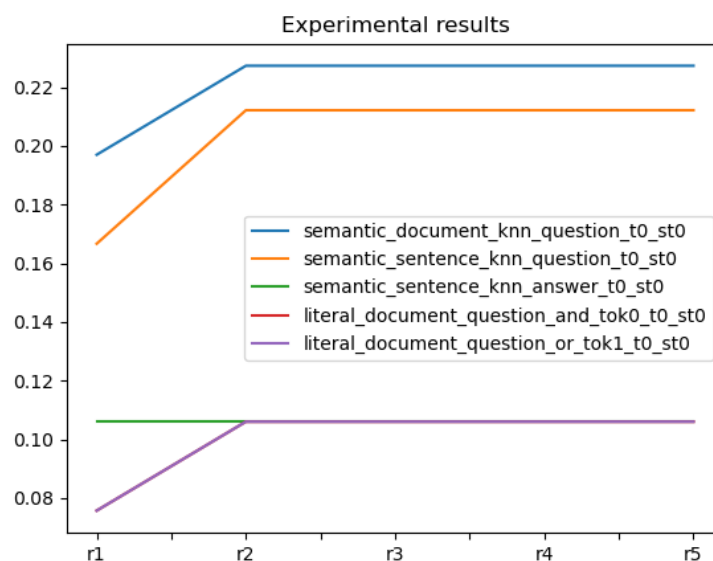


Figure 3. Results plot, showing the percentage of questions per answer rank.

We discovered that the correct answer is always on the first two options when the method provides an answer (i.e., not an empty result). In addition, semantic search can find answers to more queries than the literal search. The problem is that semantic search only answers 23% of the questions, and literal search to 11%. Looking at the results, we discovered that in many cases, Course A learning resources do not contain the answers in the text but are referenced as external documents (such as EPC guidelines or articles). Thus, external links should be indexed in order to improve search quality. This indexation was out of the scope of the project.

3.4.3. Analyze best-suited approaches to find answers to questions related to learning resources

Question-answering using LLMs involves leveraging advanced neural network architectures to generate responses to user queries. The first explored approach was to apply a two steps pipeline:

- 1) Semantically search documents with the approach described in Section 3.4.2 using the question as the search query.
- 2) Search the start and end tokens in the document that maximize the probability of being the start and end words of the answer. The text between those two words is returned as the answer to the question. This process highlights the part of the course material containing the response. However, answers are not natural. We tested using the T5 (Raffel et al., 2020) model as a summarization approach, allowing us to generate a more natural answer.

We have also experimented with auto-regressive models such as OpenAI's GPT (Radford et al., 2018) and GPT-Neo (open alternatives of proprietary GPT models) (Black et al., 2021). We have discarded big open models such GPT2 or GPT-NeoX, which require GPU with a large amount of dedicated RAM (48Gb in the case of the tested version of GPT-NeoX).

Such models have been trained with huge amounts of public (and private) information, which includes EPO website or related (as we assumed from obtained answers). We tuned such models with Course A documents to narrow their answers. The main issues observed with such models were the possibility of obtaining false information, rude language, racist comments, and many other scenarios that prevent from being used as a tool for students. However, we considered relevant to explore the possibilities of such models, including their possible benefits for teachers or as part of the training process.

3.4.4. Design a semantic search engine

After analyzing the better encodings for document search in Sections 3.4.2 and 3.4.3, we designed a search engine system to implement the different models with a reasonable amount of resources. One key issue for search engines is the required time to compute the distances, which can potentially increase with the number of documents in the corpus.

We have decided to base the implementation on Elastic Search (<https://www.elastic.co/>), an open-source search engine that implements optimal distance computation. We first indexed all the information into Elastic Search, using an appropriate representation as shown in Figure 4:

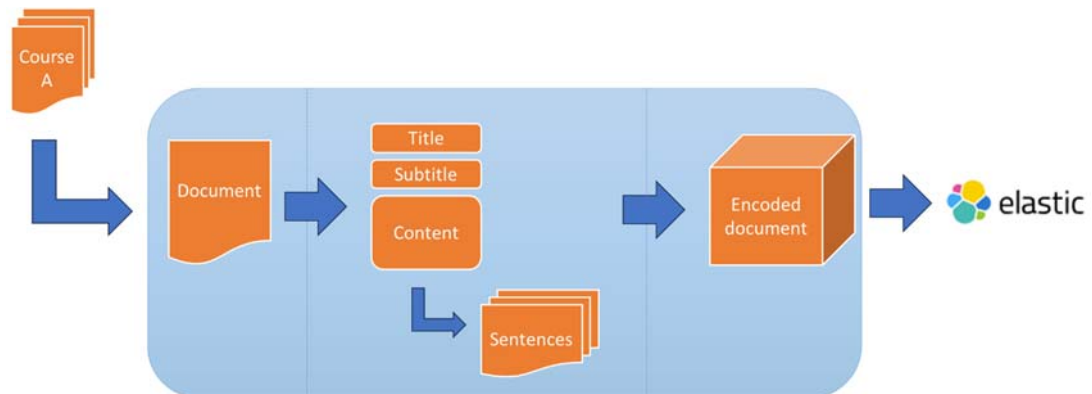


Figure 4. Representation of the indexing phase.

During indexing, each document is processed to extract the textual content using the web crawler (see section 3.4.1). The content is analyzed using an LLM to extract its sentences. All this information is then encoded using the natural language model (i.e., BERT) to generate the semantic vectors. In the case of the content, we use a model trained to be used with dot product-based distance, while the title, subtitle and individual sentences are encoded using a version of the model prepared to use cosine distance. Additionally, the index also encodes the keywords computed by the MMR approach. The final index is stored in Elastic Search.

Once all the documents are indexed, they can be searched by providing a search query or question and search options (i.e., semantic or literal search, number of responses, among others). The result is a list of documents sorted by distance (see Figure 5).

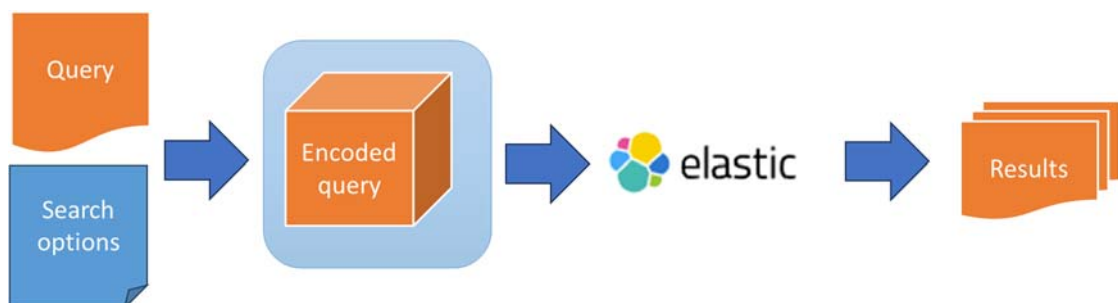


Figure 5. Representation of the search phase.

3.5. Short-answer questions assessment recommendation

The research project's next step focuses on Course A's assessment strategies. Currently, many exercises in course A are related to short-answer questions where students need to evaluate claims or sections of patents and justify whether the contents meet the patent proposal requirements.

Currently, teachers assess students' answers offline without any LMS support. Thus, the project proposed:

- To add Moodle LMS to track students' performance. Most individual exercises can be added to Moodle, and some can be automatically graded by Moodle quiz support (i.e., multi-choice, true/false, matching, drag-and-drop).

- To design a short-answer question recommender to propose an assessment and provide personalized feedback. The tool assesses students' answers and proposes an assessment to the teachers. Then, teachers review the recommendations and decide on the final score. The approach increases teachers' efficiency during the assessment process and feedback provision.

This part of the project deals with OT2. *To specify and design an integrated framework to track exercises on Course A.* The results obtained have been disseminated at Conference EARLI SIG1 2024 Assessment and Inclusivity in the Era of Digital Transformation, and summarized in Deliverables 3 and 4.

3.5.1. Analyze best-suited approaches

This research topic aims to find a technique to recommend whether a student's textual answer is correct. Similar to the search engine topic, LLMs must be used to interpret sentence meaning. BERT language model was also used as a base model since it can be run on a limited resources environment (i.e., RAM less than 32MB).

Figure 6 shows the two explored approaches and the used evaluation metrics:

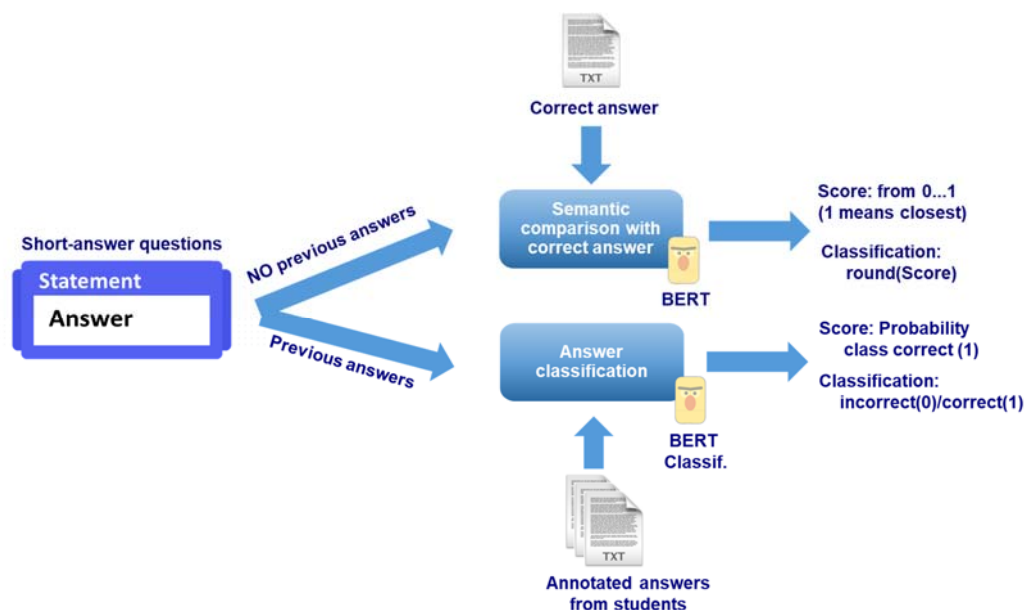


Figure 6. Different approaches for automatic assessment

- Semantic comparison with the correct answer: This approach uses BERT to create the embeddings and the cosine similarity metric to evaluate how similar the sentences are. The output of the comparison is a similarity score from 0 to 1, where 1 means the closest meaning to the correct answer. The prediction classification is obtained by rounding the score to an integer value.
- Binary classification: This approach uses previous students' answers to create a binary classification predictive model. Such answers were extracted from real students and were annotated as correct or incorrect by Course A teachers. A variant of the BERT model was used to perform binary classification and fine-tuned with such information. The model's output is the prediction of the sentence class (i.e., 0 means incorrect, and 1 means correct). In this case, the score is obtained by getting the probability of predicting the correct class (i.e., 1).

Experimental results

In order to analyze which approach is better, the Course A teachers provided eight short-answer questions with real answers from students for Clarity concept. Table 1 and Table 2 summarize the results for each approach:

Exercise	N. Correct	N. Incorrect	ACC	TPR	TNR	F _{1.5}
Claim 1	36	107	35%	100%	14%	44%
Claim 2	47	7	67%	68%	57%	78%
Claim 3	3	46	31%	100%	26%	15%
Claim 4	34	17	27%	0%	82%	0%
Claim 5	67	100	55%	98%	19%	67%
Claim 6	82	36	87%	97%	58%	91%
Claim 7	33	33	55%	81%	30%	64%
Claim 8	68	19	85%	92%	59%	90%

Table 1. Performance metrics of the short-answer questions on semantic comparison.

Exercise	N. Correct	N. Incorrect	ACC	TPR	TNR	F _{1.5}
Claim 1	36	107	83%	82%	86%	72%
Claim 2	47	7	100%	100%	100%	100%
Claim 3	3	46	90%	0%	100%	0%
Claim 4	34	17	64%	85%	25%	75%
Claim 5	67	100	81%	69%	90%	75%
Claim 6	82	36	91%	100%	66%	94%
Claim 7	33	33	77%	66%	86%	73%
Claim 8	68	19	82%	100%	25%	90%

Table 2. Performance metrics of the short-answer questions on binary classification.

Tables describe the number of annotated answers from real students, accuracy (i.e., ACC), true positive rate - TPR (i.e., correctly predicted correct answers), true negative rate - TNR (correctly predicted incorrect answers), and F-score F_{1.5} (a harmonic mean that weights both prediction classes). Binary classification models were fine-tuned by taking 80% of the students' answers. The remaining 20% of the answers were used for testing and obtaining the results of the previous table. The selection was made considering correct and incorrect answers in both datasets. Note that this training process was applied to create a specific model for each question.

Binary classification outperformed semantic comparison because students' answers provide different examples of correct and incorrect answers. Semantic search commonly fails when there are multiple ways to answer a question, and they are semantically far from the correct answer. Focusing on binary classification to detect correct and incorrect answers (i.e., TPR and TNR), quality results depended on the number of answers used for training and the balance between both classes. Models and classes with few answers are difficult to predict.

Individual analysis of wrong predictions was also performed. Figure 7 shows the analysis for Claim 5. Cosine similarity and probability of predicting the correct class for binary classification are plotted. Different colours were used to identify which predictions are

correct using both metrics (green), which ones cosine similarity fails (purple), which ones binary classification fails (yellow), and which ones both metrics fail (red).

Most errors appear on semantic comparison. Many students' answers are considered semantically similar to the correct answer provided by the teachers, ranging from 0.50 to 0.85 scores, but are incorrect. Thus, slight changes in the sentence's semantic meaning can generate incorrect predictions. Binary classification also has some erroneous predictions on detecting correct answers but fewer cases. Although these results can not be generalized, we consider that binary classification can lead to better recommendations for this domain.

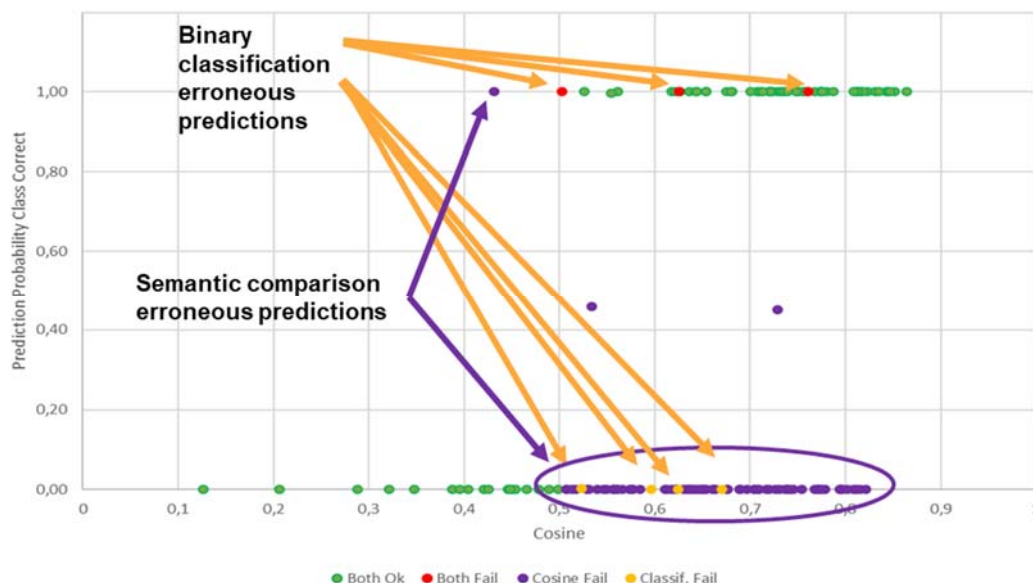


Figure 7. Accuracy comparison of the two approaches for Claim 5

3.5.2. Design a recommendation system

Based on the analysis performed in Section 3.5.1, a system was designed to assess short-answer questions. The system functionality is described next (see Figure 8):

- **Configure a question:** A new question can be added to the system by an identifying question statement. The question answer provided by the teacher, the list of annotated answers from previous students (i.e., whether they are correct or incorrect), and additional feedback are provided to configure the question. Note that when no annotated answers exist, the used metric is set to semantic comparison. The feedback provides information about the learning resources from course A, guidelines, and EPC articles that the student can review in case of submitting an incorrect answer. Finally, different configuration options are supported to set up the question. The used metric (i.e., semantic or binary classification), which type of feedback will be returned (i.e., correct answer, feedback with link to learning resources or both), and whether explainable info is shown can be configured. In the case of binary classification, the model is automatically trained for the question.
- **Send answer:** After configuring the question, answers can be sent to assess its correctness.

- Provide assessment: The system provides an assessment recommendation, generates feedback, and provides explainable information depending on the configuration options.

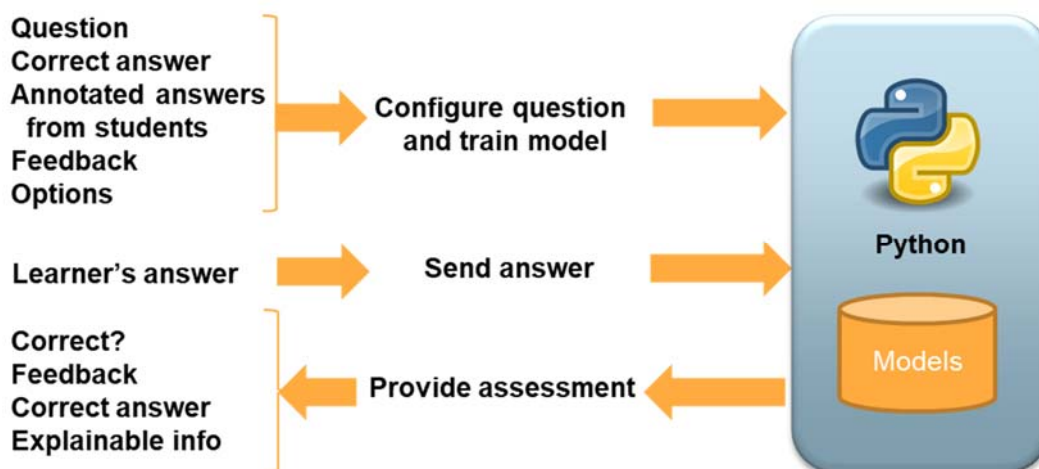


Figure 8. Assessment system operations

The system has been developed in Python. As relevant packages, Hugging Face Transformers sentence transformers (<https://huggingface.co/sentence-transformers>) were used to download pre-trained BERT classification model, and Captum package (<https://captum.ai/>) is used to visualize explainable information as shown in Figure 9 in case of the binary classifier approach:

Legend: ■ Negative □ Neutral ■ Positive			Attribution Score	Word Importance
True Label	Predicted Label	Attribution Label		
1	1 (1.00)	The claim is unclear and vague since the expression "in a" leaves it unclear whether protection for the telephone apparatus as such or the "dial tone detector".	0.08	[CLS] the claim is unclear and va ##ge since the expression " in a " leaves it unclear whether protection for the telephone apparatus as such or the " dial tone det ##et ##ctor ". [SEP]

Figure 9. Explainable information about the prediction performed by the binary classifier.

Experimental results

The recommender system has been tested with students in different editions of Course A. Two quizzes have been created in Moodle LMS with eight short-answer questions. The questions are related to the Novelty and Clarity concepts. Depending on the quiz, the students must evaluate whether a claim follows the guidelines to consider it novel and clear. If it does not, the student must justify why.

The answers are sent to the recommender system, and the teacher has to check whether the recommendation is correct and provide the student with the proper grade. The experiment focuses on testing the quality of the recommendations after the teacher's revision. Specifically, two different scenarios have been tested:

- Novelty Quiz: The quiz was tested in two editions of Course A. Previous students' answers were unavailable. Thus, the first edition used the recommender with semantic comparison. The second edition used the answers graded (and annotated whether the answers were correct) from the first to train the binary classification models. This setting aims to compare the accuracy of both approaches.
- Clarity Quiz: The quiz was tested in four editions and there were annotated previous answers from the beginning. Thus, binary classification was used in all editions. Additionally, models were retrained on each edition by adding new answers graded from previous editions. This scenario aims to check whether there is any improvement in retraining the models with new data.

Aggregated results are shown in Figure 10 for Novelty quiz. The results of all questions have been aggregated to simplify the evaluation. We can observe a significant difference between both approaches. Global accuracy (i.e., ACC) improves from 48.66% to 62.01%. The meaningful increment in detecting wrong answers caused this improvement (i.e., the TNR increased from 32.81% to 71.88%). However, the detection of correct answers decreases when binary classification is used. Similar to the experimental results in Section 3.5.1, semantic comparison tends to recommend that most students' answers are correct. This produces the unwanted counter effect that semantic comparison cannot correctly detect most incorrect answers. Thus, changing from semantic comparison to binary classification when there is some annotated information positively impacts the recommender.

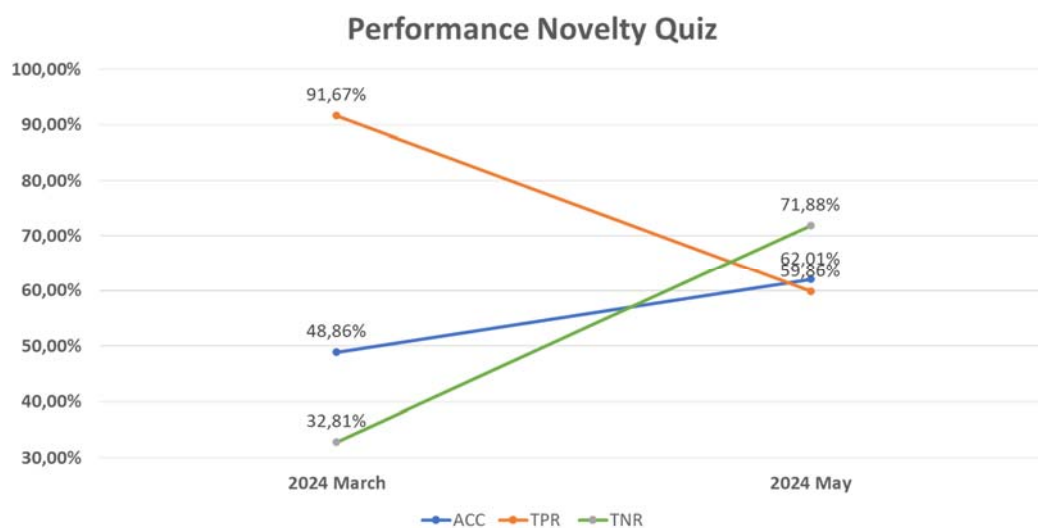


Figure 10. Performance comparison on Novelty quiz.

The results of the second experiment are illustrated in Figure 11. Results have also been aggregated. We can observe that retraining the binary classification models is positive for the recommender's accuracy. Detection of wrong answers (i.e., TNR) increases during all editions until reaching an accuracy of 88.53%, and the detection of correct answers (i.e., TPR) finally increases in the last edition to 95.56%. The relevant insight in this experiment is that in few editions, the binary classification models are capable of producing high-quality recommendations.

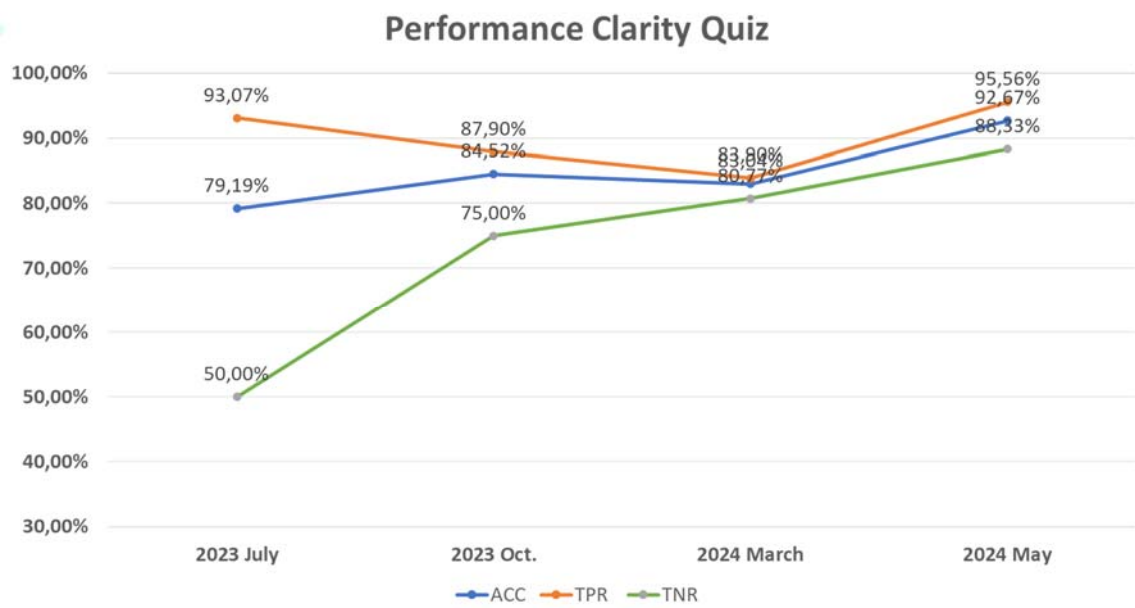


Figure 11. Performance comparison on Clarity quiz.

4. Patent Academy

4.1. Preliminaries

4.1.1. Course definitions

The Patent Academy is focused on standalone e-learning courses with many different characteristics that enrich the learning process that can be applied. The courses are performed with Moodle LMS, allowing tracking of students' actions and performance. Additionally, such information can be used to create predictive models and recommenders to support students during their learning process.

The Patent Academy has different types of courses:

- *Guest courses*: General information about one topic. They are free (at no cost). This type of course does not require to be registered on the platform.
- *Free courses*: These courses may have (or not) an assessment at the end. They are more complete (more resources) compared with the previous ones. They can provide a certificate at the end when the final quiz is passed with a minimal grade. The student must be registered on the platform to perform this type of course and, therefore, obtain the completion certificate (when available).
- *Paid courses*: The educational distinction between paid and free courses is unclear. We assume they are more complete and have a teacher; thus, students pay for the certificate or the teacher's support. They may include assignments that teachers assess. They may not be graded. Teachers may only check weak points (feedback). The course may include live sessions. They may or may not have a quiz at the end. Some courses are similar to webinars, with a live session every month/week.
- *Restricted courses*. These courses are managed internally and are similar to paid courses. Students are registered manually to access the resources, activities, and certificates.

Based on this description and the Moodle course structure, we identified different types of courses:

- *Course without activities*: This course may have learning resources, web links, live sessions, and forums but without any activity, i.e., quizzes or assignments.
- *Course with ungraded activities*: This type differs from the previous one in that there are assignments that are not graded with a score.
- *Course with graded activities*: In this case, the activities (quizzes or assignments) are graded.

Additionally, these types can be complemented with the definition of completion since there can be courses (i.e., with no ending date) where no event indicates that the student has finished the course. Thus, we have defined different ending events:

- *Passed event*: A course is passed when the student has a grade or an ending certificate.
- *Completed event*: A course is completed when the student has a Passed event or submitted all the assignments/quizzes (being graded or ungraded).

- *No ending event*: The course is similar to a resource material course to review without any ending event (i.e., no grade and no certificate), and there are no activities to check the Passed or Completed event.

4.1.2. Predictive outcomes

Four predictive outcomes were identified:

- *Predicting students' failure*: The objective is to predict whether a student will pass/complete a course based on assignment/quiz information.
- *Predicting students' dropout*: The objective is to predict the time of dropping out of the course (in days) based on students' interaction with the Moodle course. Note that this event is slightly different from the previous one. A student who drops out will fail, but not inversely.
- *Recommending the next activity to perform*: Although Moodle can be configured to perform the course and activities sequentially, students may decide to do the activities in any order. This may happen in courses with a large set of quizzes to practice. The objective is to recommend the next quiz to perform.
- *Recommend the next enrolled course*: Based on enrollment history, the objective is to recommend a list of the next courses to enrol in.

Table 3 shows the potential applicability of models on the type of courses based on their definitions. The final application is limited to each course's configuration and available data. Predicting students' failure can only be applied when there are activities in the course and an ending event. Note that the model can also work in courses with ungraded activities (in such cases, the activity submission event is used instead of the activity grade). Next activity recommendation is also limited to courses with activities. Predicting students' dropout can be applied in all courses since students' actions are used to predict the time to drop out. Enrollment recommendations can also be applied to all courses since enrollment information is used.

	No ending			Completed			Passed		
Predictive Model	No Act.	Act. without grade	Act. with grade	No Act.	Act. without grade	Act. with grade	No Act.	Act. without grade	Act. with grade
Predicting students failure					✓	✓		✓	✓
Predicting students dropout	✓	✓	✓	✓	✓	✓	✓	✓	✓
Recommending next activity		✓	✓		✓	✓		✓	✓
Recommending course enrollment	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 3. Predictive models application depending on the type of course.

4.1.3. Data model

Before creating any predictive model, we need to know the available data. This project assumes the Moodle LMS and the data model in version 3.9.1. Although the Moodle version is currently higher, the changes in the database are minimal in terms of the data used in this project.

Moodle data used are described next:

- mdl_user_enrolments: Information about enrollment (all courses).
- mdl_course: Information about a course.
- mdl_course_completions: Information about course completion (all courses).
- mdl_course_completion_crit_compl: Information about the final grade and time spent completing the course (it depends on course configuration).
- mdl_course_modules: Relationship between course and modules.
- mdl_course_modules_completion: information about modules/section completion (it depends on course configuration).
- mdl_course_sections: Relationship between course and sections.
- mdl_assignment: Information about course assignments (legacy support to old course versions).
- mdl_assignment_submissions: Information about assignment submission (legacy support to old course versions).
- mdl_assign: Information about course assignments.
- mdl_assign_grades: Information about the grade of an assignment when they are graded with a score.
- mdl_assign_submission: Information about assignment submission.
- mdl_quiz: Information about a quiz.
- mdl_quiz_grades: Information about the grade of quizzes when they are graded with a score.
- mdl_forum: Information about course forums.
- mdl_forum_discussions: Relationship between forums and courses.
- mdl_forum_posts: Information about forums write interaction.
- mdl_forum_read: Information about forums read interaction.
- mdl_chat_messages: Information about quick messaging service among students.
- mdl_logstore_standard_log: log information about all the actions/events performed for users (students/teachers) on courses (it depends on course configuration).

Although data can be used straightforwardly, we decided to extract and transform it into a new simplified data model. Additionally, user data has been anonymized during this process to avoid privacy issues. The new data model is described next:

- ENROLMENT_USER_SUBJECT: Information about the enrollment of a course.
- CLASSROOM_USER_PASSED: Information about passed courses.
- CLASSROOM_USER_COMPLETED: Information about completed courses.
- CLASSROOM_USER_ACTIVITY_SUBMIT: Information about submitted assignments/quiz.
- CLASSROOM_USER_ACTIVITY_ASSESSED: Information about the score of an assignment/quiz.
- USER_MESSAGE_READ: Information about reading a message in a forum/chat.
- USER_MESSAGE_WRITE: Information about writing a message in a forum/chat.
- SUBJECT_INFORMATION: Information about a course.

4.2. Objectives

This part of the project focuses on using Moodle LMS data to create AI predictive models to support students' learning process. The project explores all the phases of developing a predictive analytics system, from gathering the data, training the models, building the EWS, and evaluating the predictive models' performance and identifying at-risk events.

The aim of the project for Patent Academy is:

- OP1. To analyze the available Moodle data.
- OP2. To specify and design an Extract-Transform-Load (ETL) process to extract data from the Moodle database.
- OP3. To specify and design predictive models for predicting students' completion.
- OP4. To specify and design a predictive model for recommending the next course to enroll in.
- OP5. To specify and design a web interface for teachers to visualize such data.
- OP6. To specify and design a recommendation system to inform different stakeholders (i.e., teachers and students) of the different outputs of the models.

The experiments were conducted on different datasets created from Patent Academy Moodle LMS. For each course, the data has been split into three datasets: 1) the training dataset, 2) the validation dataset, and 3) the testing dataset. The training dataset was used to train the models, and the validation one was used to validate the trained models. The testing dataset was used to simulate an independent set of students using the developed EWS system and evaluate the at-risk identification.

4.3. An ETL process to extract and anonymize Moodle data

This section deals with objectives *OP1. To analyze the available Moodle data* and *OP2. To specify and design an Extract-Transform-Load (ETL) process to extract data from the Moodle database*. An ETL process was designed to extract data from the Moodle database.

Technically, the ETL was designed in Python to read the Moodle database and extract meaningful data. The extraction also anonymizes the user identity by replacing it with an alphanumeric random hash of 64 digits. This implementation decision was made to maintain anonymous students' identities during the research project. The relationship between the random hash and the username is stored securely in a Redis database that EPO kept internally. This relationship will be used if each student's identity and predictive data need to be matched.

The extraction system has been designed to perform a full or incremental extraction. The second option stands for an extraction operation on a production environment where the data for Moodle is extracted to perform daily predictions. In this project, only the full extraction was used to obtain data to create the different datasets for training, validating and testing.

The process extracted data related to enrollment, subject performance, forum, module completion, quiz submission, and quiz grading as described in Section 4.1.3. Moodle logging information was also analyzed to extract clickstream data. The process extracted 387645 registries on a total of 253 courses. However, the data were cleaned by removing courses with too few students (i.e., less than 30 enrollments) or meaningless data (i.e.,

courses with no data). The number of registries after the cleaning process was 231544 on 103 courses.

Some of those courses were different editions of the same course. Therefore, the analytical process identified 64 unique courses. The total data is summarized in Table 4.

	Raw data	After cleaning
N. courses	253	103
N. enrollments	27558	25619
N. course performance	23174	23170
N. forum read	23054	19232
N. forum write	13804	13374
N. modules completion	12216	11636
N. quiz grading	75046	43550
N. quiz submission	119873	74864
N. clickstream	92667	19996
Total registries	387645	231544

Table 4. Extracted and cleaned data from Moodle database.

4.4. Detect at-risk students of failing a course

This section focuses on objectives *OP3. To specify and design predictive models for predicting students' completion*, *OP5. To specify and design a web interface for teachers to visualize such data*, and *OP6. To specify and design a recommendation system to inform different stakeholders (i.e., teachers and students) of the different outputs of the models*.

Different predictive models have been explored to predict students' failure. These models are validated with the validating dataset to check their accuracy. As a second output, the predictive model has been integrated into the EWS to identify students with a potential risk of failure based on the testing dataset. The results have been summarized in Deliverables 5 and 6.

4.4.1. PGAR failure model

The objective outcome of this model is to predict whether a student will fail a course. We define *fail* as the student is not completing or passing the course. Note that the completing or passing event is selected depending on which ending event has the corresponding course in Moodle.

The predictive model uses the PGAR (Profiled Gradual At-Risk) model as a base model (Baneres et al., 2020) built for each course and composed of a set of predictive models defined as submodels. A course has a submodel for each assessment activity. Each submodel has features such as the student's profile information and the information about the activities (i.e., submission or grade) until the current one. The model outcome is to *fail* the course, and it is a binary variable (i.e., fail or pass/completed).

Example 1. Let us describe the PGAR model for a course with four activities (AA). In such a case, the PGAR model contains four submodels:

$$\begin{aligned} \text{PrAA1(Fail?)} &= (\text{Profile}, \text{InfoAA1}) \\ \text{PrAA2(Fail?)} &= (\text{Profile}, \text{InfoAA1}, \text{InfoAA2}) \\ \text{PrAA3(Fail?)} &= (\text{Profile}, \text{InfoAA1}, \text{InfoAA2}, \text{InfoAA3}) \\ \text{PrAA4(Fail?)} &= (\text{Profile}, \text{InfoAA1}, \text{InfoAA2}, \text{InfoAA3}, \text{InfoAA4}) \end{aligned}$$

where $\text{PrAA}n(\text{Fail?})$ denotes the name of the submodel to predict whether the student will fail the course after the activity $\text{AA}n$. Each submodel $\text{PrAA}n(\text{Fail?})$ uses the student's profile (*Profile*) and information about the activities (*InfoAA1*, *InfoAA2*, ..., *InfoAA* n). The number of previously enrolled courses, the number of passed/completed courses, and the number of times the student has repeated the predicted course are used as profile information. The information about the activity (*InfoAA* n) uses the grades or the submission event from the first activity until the activity $\text{AA}n$.

Since not all courses have grading information related to activities and some courses do not have pass information, four models have been explored:

- Predicting failure based on passed event with activity grading information.
- Predicting failure based on completed event with activity grading information.
- Predicting failure based on passed event with activity submission information.
- Predicting failure based on completed event with activity submission information.

Experimental results

The experimental results are computed based on the training and validating dataset. In this case, the training dataset was used to train the model based on the submission/grade of the activities and the outcome Fail or Pass/Completed. The validating dataset was used without the known outcome (Fail or Pass/Completed) to check whether the model can predict it.

The experimental results are reported in Table 5 where accuracy, MAE, TPR, TNR and $F_{1.5}$ are summarized.

Predictive Model	N. Courses	ACC	MAE	TPR	TNR	$F_{1.5}$
Pass by Grade	22/64	92.20%	0.07	71.28%	92.51%	93.64%
Pass by Submit	23/64	85.55%	0.14	80.70%	85.85%	88.44%
Completed by Grade	22/64	92.56%	0.07	94.49%	92.90%	92.85%
Completed by Submit	23/64	88.96%	0.11	93.52%	88.63%	89.34%

Table 5. Failure models performance summarization.

After analyzing the results, we can conclude the following insights:

- The models are highly accurate on average in terms of accuracy (i.e., predicting either failing or passing/completing) and detecting correctly passed/completed events (TNR).

- When analyzing detecting failure (TPR), the model fails in some courses because students tend to pass/complete the courses. Thus, models are trained on unbalanced data, making failure prediction more difficult to detect.
- These models can only be applied to courses with assignments or quizzes.
- Models, where quizzes or assignments are graded (with a score), tend to perform better because a score has more information about the knowledge acquired.
- These different models were explored due to the variability of the available data on courses. Then, when applying these models in the EWS, the system can select the best one depending on available data, performance, or teacher criteria.

4.4.2. At-risk failure identification

The failure identification is at the activity level. Therefore, if a course has 4 activities, the failure risk is analyzed 4 times. The submodels presented in the previous section produce a binary prediction: fail or pass/completed. Given that such a prediction brings poor information to a student or teacher, this information is transformed into a prediction of the minimum grade that must be obtained in the current activity to be able to pass the course. Note that, in the case of using the submission event, the prediction focuses on informing whether the submission of the activity impacts on passing the course. From now on, we will refer to the grading case. However, it is straightforward to change to the submission case.

The at-risk level, which is a prediction, is displayed in a three-colour at-risk level semaphore (green, yellow and red) that informs students of the risk associated with each grade they can obtain in the current activity. It is worth noting that this information can also be presented to the student. The current activity can be predicted once the previous one has been assessed (or, in the case of the first activity, it is shown at the beginning of the course). Therefore, it gives relevant information about the minimum grade the student needs to obtain to follow the course successfully. Therefore, each student can self-regulate their effort to get the minimum qualification that informs the at-risk semaphore.

Figure 12 shows a hypothetical case. The student is informed that if he/she obtains an A in the current activity, he/she will probably pass the course with the grades he/she has obtained so far. Not submitting the activity (grade N) or failing (grades F or D) will negatively impact his chances of passing. It also informs that a C or B grade may imply failure depending on the grades obtained in future activities.



Figure 12. Failure at-risk semaphore before grading the activity

Once the activity has been submitted and assessed, this semaphore is updated (see Figure 13), indicating the student's at-risk level (i.e., the grade the student has finally obtained is marked informing about its current associated risk). Therefore, this semaphore has two objectives: 1) informing the potential at-risk levels before submitting the activity, and 2) reporting the at-risk level once the activity has been assessed.

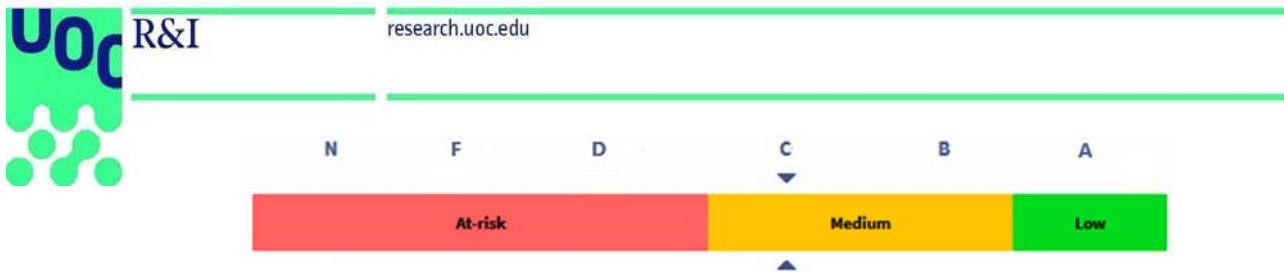


Figure 13. Failure at-risk semaphore after grading the activity

However, it is relevant to know how the risk associated with each grade is calculated. This calculation is performed with a simulation using the corresponding predictive submodel to the activity to find which grade changes the prediction from failure to pass.

For example, suppose we have the predictive submodel for the first activity of a course, and we want to know the minimum grade that a student must obtain according to the corresponding submodel associated with the first activity. Six simulations were conducted using the different grades the student can obtain in the activity and the student's profile (*Profile*). Each simulation will predict the probability of failing according to the indicated grade. Here is a hypothetical example:

- Model Act1 = (Profile, N) Student will fail? = Yes
- Model Act1 = (Profile, F) Student will fail? = Yes
- Model Act1 = (Profile, D) Student will fail? = Yes
- Model Act1 = (Profile, C) Student will fail? = Yes
- Model Act1 = (Profile, B) Student will fail? = No
- Model Act1 = (Profile, A) Student will fail? = No

We can observe that the student with the profile *Profile* has a high probability of failing the course if he/she obtains a grade below B. In other words, students who obtain a grade of N, F, D, or C in Activity 1 with the assumed profile will tend to fail the course. However, it is perfectly possible to pass with these grades, but less often, according to the predictive submodel, that is not totally accurate.

Finally, the at-risk level of failing the course is computed based on a decision tree shown in Figure 14, which classifies students into different statuses. Such statuses are used to send specific messages when the platform is used as an intervention system. Green is assigned when the grade is higher than the predicted minimum grade, and two different statuses are identified based on the obtained grade. Outstanding students (i.e., grades larger or equal to B) can be congratulated, while the message for others can be more neutral. Note that a student can obtain a D grade and be predicted as green (i.e. for non-mandatory activities), and the message can not have the same intensity as a message for a student who obtained an A. Red is assigned when the student fails the activity, and Yellow when the activity is passed, but the grade is lower than the minimum predicted grade. Students who do not submit the activity are classified as another at-risk level (i.e., black colour) depending on the number of non-submitted consecutive activities.

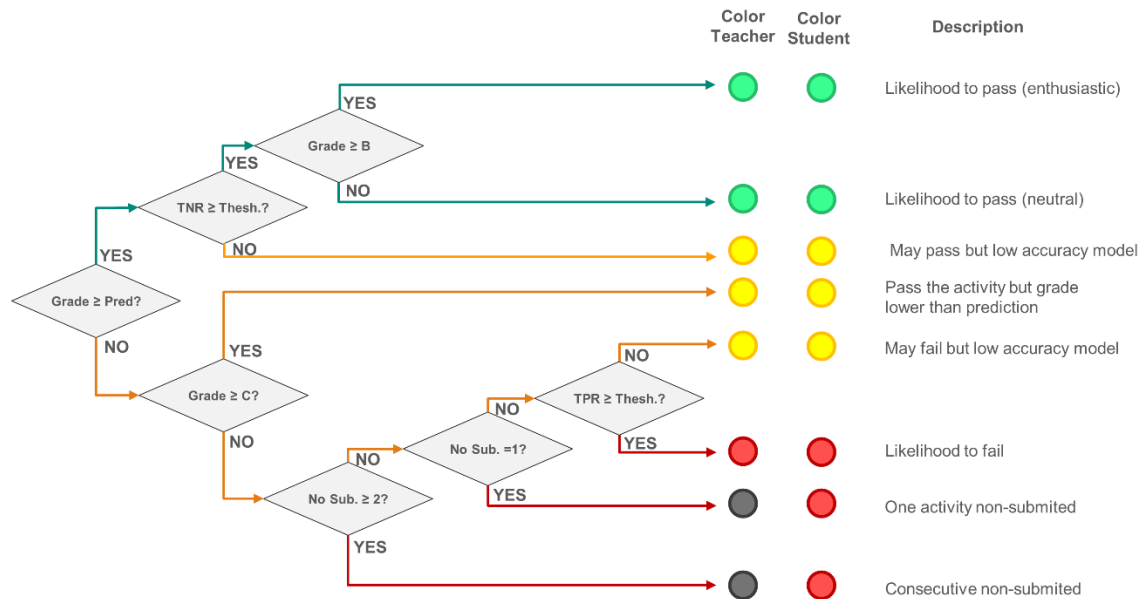


Figure 14. Decision tree for the at-risk failure identification

The at-risk level classification also considers the submodels' accuracy in detecting at-risk students (true positive rate - TPR) and non-at-risk students (true negative rate - TNR). At-risk levels are strictly assigned when high-quality submodels are used. However, at-risk levels cannot be guaranteed on low-quality submodels. In such cases, the at-risk level is reduced to a medium at-risk level (i.e., yellow). Submodels are denoted as high-quality when their accuracy (i.e., TNR or TPR) is above a specific threshold. The threshold is set to 75% by default but can be modified within the Slasys platform.

Experimental Results

For each course, the different failure predictive models have been trained, and the best one in terms of accuracy has been selected. Then, a simulation has been performed using the testing dataset. The simulation's performance was calculated using the same metrics as for the training process (i.e., accuracy, TPR—correct identification of at-risk students, TNR—correct identification of non-at-risk, and F-score). The final performance was an accuracy of 86.11%, a TPR of 96.97%, a TNR of 64.48%, and an $F_{1.5}$ of 90.93% for 18 courses.

We observed that failure prediction could be possible for more courses since information related to activities was available. However, in some courses, all students submitted or passed the activities, making the training process impossible (i.e., predictive models need information about submitting/not submitting or passing/failing to perform the training operation).

The analysis of the results concluded:

- The global accuracy is higher than 85%. Only one course had an accuracy lower than 60%.
- The identification of at-risk students is highly accurate. All courses have a TPR higher than 90% except one course.

- The identification of non-at-risk students was significantly lower. However, 12 courses (of 18) have a TNR larger than 50%.

Note that these results should be examined with caution since they are based on the extracted data from Moodle. Since analyzing the export process was not possible, the results might be inaccurate. Additionally, the models' accuracy is affected by the captured ending events. Since some courses do not have such an event, it was inferred from the log table when the student accessed the certificate or the day the course's final score was generated.

4.5. Detect at-risk students of dropping out of a course

This section also focuses on the objectives of the previous one (i.e., *OP3*, *OP5*, *OP6*). However, only one predictive model has been explored to predict students' dropout. The model has also been validated with the validating dataset to check their accuracy. Similarly, the predictive model has been integrated into the EWS to identify students with a potential risk of dropping out based on the testing dataset, and the results have been summarized in Deliverables 5 and 6.

4.5.1. WTTE dropout Model

Aforementioned, dropout has different meanings in education, depending on the context. On the highest level, dropout materializes when a student stops enrolling in an education system or institution. On a middle level, dropout is materialized when a student stops accessing an enrolled course. On the lowest level, when a course is divided into different topics or categories, dropout materializes when a student does not reach the next topic or category.

This project uses the middle-level dropout definition to determine when a student will stop accessing the course. The *dropout* is materialized when a student stops accessing an enrolled course. This prediction could be done with a classical classification algorithm giving a binary outcome (dropout / continue on course). However, based on available data, this approach leads to low-accurate models.

We transformed the prediction outcome into a survival question. Instead of asking if (today) the student will drop out of the course, which produces a binary outcome, the model asks how many days remain until the student will drop out of the course. This question is answered with a number of days, and reaching a value near zero means that the student will probably drop out in the following days.

Survival analysis is a branch of statistics designed to analyze the expected duration until an event of interest occurs. There are different models for survival analysis, but in this project, we explore Weibull time-to-failure model (WTTE) (Ho & Silva, 2006). This model is based on Weibull distribution (Weibull, 1951), which models the nature of a time to failure or the time between two events. Thus, the WTTE model predicts the time until failure or when the next event will probably happen.

The dataset structure differs from the needed one for classical classification models since the concept of time should be included within the data. Assuming a user u , a course c and a day t , a row of the dataset contains:

- User ID: Identification of the student u .
- Num. Enrolled: Number of enrolled courses.
- Num. Repeated: Number of times enrolled in course c (repeated).
- Num. Passed: Number of passed courses.
- Num. Completed: Number of completed courses.
- Average Access Course: Average number of accesses of user u to course c from the starting date until day t .
- Average Reading Forum: Average number of forum reads of user u to course c from the starting date until day t .
- Average Writing Forum: Average number of forum writes of user u to course c from the starting date until day t .
- Submit Activity 1?: Whether user u has submitted activity 1 on course c until day t .
- Submit Activity 2?: Whether user u has submitted activity 2 on course c until day t .
- ...
- Submit Activity n ?: Whether user u has submitted activity n on course c until day t .

where it is used the student's profile as for the failure model described in Section 4.4.1, click stream data (i.e., data about utilization of the LMS), and performance data (i.e., activities submission). The dataset will have a row for each tuple (user u , course c , day t) storing the student's profile, whether the student is engaged in the course, and his/her performance progression. Additionally, using average values on click stream data gives the model knowledge about how active the student is over time.

Although the predictive model mostly correctly detects students who can potentially drop out, false positives (i.e., erroneous dropout students detected) are also observed because the students are not always active in Moodle. We propose to define a confidence interval of days a student can be predicted as a potential dropout to minimize errors by using a dropout temporal window (DTWin) (Bañeres et al., 2023). This interval allows to maximize the detection accuracy. An optimization process is performed to compute the best confidence interval near the day students are predicted as dropout (i.e., the model predicts zero value).

Figure 15 shows the result of the optimization process on a course. Assuming a confidence interval of 0 days, we can observe that the accuracy of detecting dropout (i.e., TPR) is 0%. However, the optimization process finds that 5 days is the best interval since TPR increases to 68% without losing performance on non-at-risk dropout students (i.e., TNR). In other words, selecting a confidence interval of 5 days in this course suggests that when a student is predicted with a value less than 5, he/she is probably near to stop accessing the course.

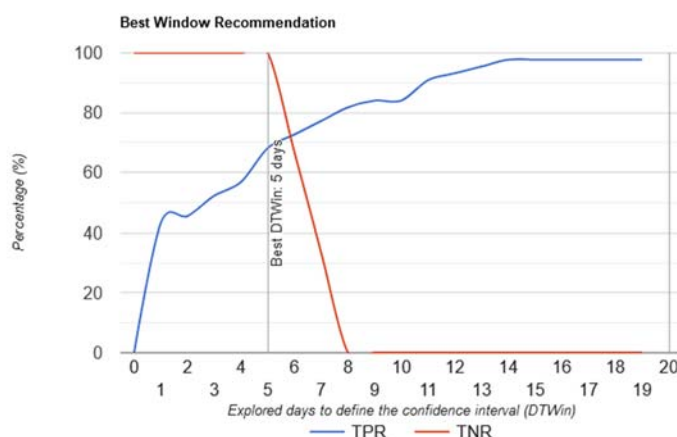


Figure 15. Confidence interval computation

Experimental results

The testing and validation datasets were also used for the experimental results. However, the training dataset was split into two (i.e., 80% and 20%) since the neural network needs an evaluation dataset during training. In the validation dataset, the registers for each student were randomly pruned. The model aims to predict the remaining days until the student stops accessing.

Aggregated results are summarized in Table 6, where accuracy, MAE and RSME are computed. Note that accuracy is computed by checking whether the predicted number is within the computed confidence interval.

	N. Courses	ACC	MAE	RSME
WTTE-RNN	33/64	72.80%	5.03	2.65

Table 6. Dropout model performance summarization.

The analysis of the results concluded:

- This model can be applied to courses without activities.
- This model highly depends on the information within the Moodle log (i.e., data about students' access and certificates issued).
- The model mispredicts on the course's initial days because there are too few registers to predict accurately.
- The model does not work for courses with students' maximum access of 10 days. The model makes sense on more extensive courses.

4.5.2. At-risk dropout identification

Similar to the failure model, the dropout prediction must be transformed to an at-risk level. A prediction is performed daily for each student, giving the number of days remaining until the student will probably stop accessing the course. Such prediction is displayed as an at-risk semaphore, but, in this case, the visualization shows the days consumed from the confidence interval.

A student consumes one day when the model predicts a value smaller than the size of the confidence interval. For instance, let us assume a course with a dropout model with a confidence interval of 4 days. A student will consume one day when the model will predict that he/she will drop out in less than 5 days (i.e., 0 to 4). Figure 16 shows that a hypothetical student has consumed 3 days of the interval. In other words, he/she has obtained 3 consecutive predictions with values from 0 to 4. In this visualization, when a student consumes all days of the interval, the dropout at-risk alarm will be raised.



Figure 16. Dropout at-risk semaphore.

Although previous semaphore visualization offers a gradient between green and red, the student has a specific at-risk level colour for each day. The same conceptualization performed for the failure model is applied by using a decision tree to assign the at-risk level (see Figure 17).

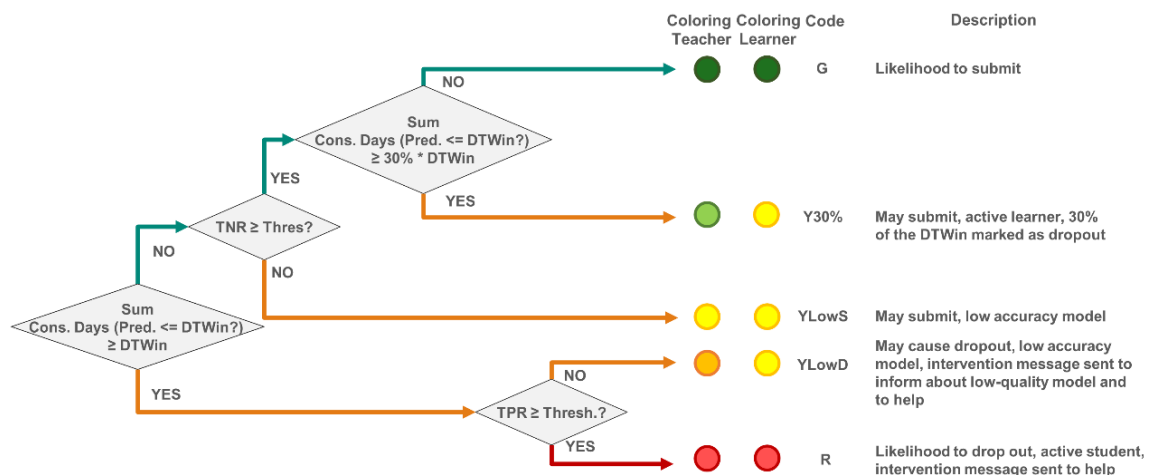


Figure 17. Decision tree for the at-risk dropout identification

The colours are different for students and teachers. While students only have three-colour at-risk levels (green, yellow and red), teachers have more colours to understand each student's status better. Specifically:

- The green colour is assigned to students with low risk and students who do not reach 30% of the confidence interval size (i.e., DTWin in the figure).
- Yellow is assigned to students at medium risk and has different meanings. This colour is given to students who have consumed more than 30% of the confidence interval. Similar to the failure model, it is also assigned to low-quality predictive models (i.e., where the accuracy of detecting at-risk students –TPR and detecting non-at-risk students –TNR are below the established threshold).
- Finally, the red colour represents a high risk of dropping out when the confidence interval size has been consumed and a high-quality model is used (i.e., with TPR above the threshold).



Note that, the threshold is set to 70% by default, but it can be modified within the Slasys platform.

After different experiments, we observed that the models started producing meaningful predictions after 5 days of data. Therefore, the first prediction starts after two days of data (i.e., access, read or submission) within a minimum interval of 5 days.

Experimental results

The accuracy of at-risk dropout identification has also been performed on the testing dataset. However, two analyses were performed depending on the condition to check when the dropout at-risk alarm was raised. The first analysis is based on whether the student has finally completed/passed the course.

The metrics used are the same as those used in the failure model (i.e., accuracy, TPR, TNR, and F-score). The final performance was an accuracy of 44.17%, a TPR of 53.75%, a TNR of 42.69%, and a F15 of 54.02%. The results are inaccurate because:

- The model's outcome is not to predict whether the student will pass/complete the course. Thus, the obtained result is comprehensive.
- A student can pass a course even if the dropout alarm is raised.
- This analysis is also affected by the same problems identified for the failure model (i.e., some ending events have not been correctly captured within Moodle LMS).

The second analysis focuses on whether the student has accessed to the course after the dropout alarm. This analysis better aligns with the model outcome since the dropout model predicts whether a student will stop accessing the course.

The only metric used is TPR. Here, the TPR is computed based on checking whether predicted at-risk students did not access the course after the alarm was raised. The final performance was a TPR of 65.34%, whereas 10 of 21 courses had a TPR smaller than the average.

The acquired data also limits the results:

- The dropout model combined with the confidence interval successfully identifies students not accessing the course for several days. Some of them will inevitably access the course after the alarm, but if it is used as an intervention system, this alarm could be used to reengage them in the course.
- The accuracy is highly affected by the lack of clickstream data in many courses (i.e., some old courses did not have logging information due to the configured clean-up Moodle policy). Due to this problem, the prediction was not possible for many students.

4.6. Next activity recommendation

Submitting activities may not follow a sequential order on courses with multiple quizzes/assignments where the student uses such activities to practice. This recommendation proposes the next activity to do within a course.

Note that this recommendation can be combined with failure models, providing early feedback on students at potential risk of failure by recommending the next activity to perform and, thus, re-engaging them in the course.

The recommender system uses the compact prediction tree (CPT) (Gueniche et al., 2013). Compared to other predictive models, its main distinctive characteristics are 1) that CPT stores a compressed representation of training sequences with no loss or a small loss, and 2) CPT measures the similarity of a sequence to the training sequences to perform a prediction. The similarity measure is noise tolerant and fast and, thus, allows CPT to predict the next items of subsequences that have not been previously seen in training sequences.

In the case of the next activity prediction, the model uses the submission order performed by the students on courses. Such an order creates the sequence of submissions performed by each student. The experimental results were reported in Deliverable 5.

Experimental results

In this case, the experimental results were performed using a training and validation dataset with the sequence of submissions ordered by date in all the courses with available data. In the testing set, we removed the last item of the sequence (i.e., the last submitted assignment/quiz), and the objective of the test was to predict this last item.

Since it is a recommender system, we predicted different k numbers of recommendations ranging from 1 to 5. The model correctly predicts when the last item removed is in the list of recommendations. The experimental results for $k = 1, 3$, and 5 are summarized in Table 7 where the Precision@ k , the MAE and RSME are shown.

Number of Recommendations	N. Courses	Precision	MAE	RSME
@K = 1	28/64	83.00%	0.17	0.27
@K = 3	28/64	94.87%	0.05	0.08
@K = 5	28/64	98.03%	0.02	0.04

Table 7. Next activity recommendation performance summarization.

The analysis of the results concluded:

- Similar to the failure model, this model can only be applied to courses with assignments or quizzes.
- The model is highly accurate on courses with few activities with no high sequence variation.
- The model accuracy decreases on course with more activities, but it can be partially compensated by increasing the number of recommendations.

- The model is highly affected when the testing course modifies the number of assignments/quizzes. Introducing new activities in the middle of the course will lead to situations when these new activities will not be recommended.

Note that there is no further analysis in the simulation step within the Slasys system since the recommendation can not be delivered to real students. The recommendation is computed during the simulation but not evaluated.

4.6. Course enrollment recommendation

The last predictive model explored was for course enrollment recommendation. This part of the project contributes to objective *OP4.To specify and design a predictive model for recommending the next course to enroll in*. The experimental results were reported in Deliverable 5.

There are two approaches to perform these recommendations:

- Collaborative filtering: It is a type of recommendation that uses the collective behaviour of users to recommend items. It works by analyzing the user's behaviour and preferences and recommending items other users with similar behaviour and preferences have also selected. The system uses the historical data of users' interactions (i.e., enrollments in our case) with items (i.e., courses in our case) and identifies patterns of similarity between users.
- Content-based filtering: It is a recommendation that uses items' attributes to recommend similar items to the user. It works by analyzing the items' properties (i.e., in our case, it could be the type of course, paid/free, among others) and recommending items with properties similar to the user's. This type of recommendation focuses on the items' features and the users' preferences rather than the behaviour of other users.

LightFM (Kula, 2015), a hybrid matrix factorization model for representing user and item relationships, has been used to perform this recommendation. LightFM combines collaborative and content-based filtering to learn relationships between users and items. However, it also considers users' metadata (i.e., gender, age, preferences) and items' metadata (i.e., type of item, pricing).

In the case of course enrollment, the enrollment information is coded as an enrollment matrix $A \in R^{m \times n}$, where m is the number of students and n is the number of courses. The element a_{ij} within the matrix is 1 when the student i has enrolled in the past in course j . Additionally, student metadata can also be described with a similar matrix $U \in R^{m \times d}$, where m is the number of students and d is the number of characteristics that define students and their preferences. The element u_{ij} within the matrix is 1 when the student i has the characteristics j . Finally, course metadata can be also defined with a matrix $C \in R^{n \times p}$, where n is the number of courses and p is the number of properties which describe the courses. The element c_{ij} within the matrix is 1 when the course i has the property j .

Given these three matrices to the LightFM algorithm, the matrix factorization is automatically built and can be used to train the enrollment recommendation model. In this experiment, the model was fed with matrix A and matrix U where d has only one characteristic related to the number of enrolled courses. However, no matrix C was added since course properties were not extracted from the Moodle database.

Based on this information, four experiments were conducted:

- Enrollment information matrix A without students' metadata matrix U for only considering course relevancy in terms of enrollment.
- Enrollment information matrix A with students' metadata matrix U for considering course relevancy in terms of enrollment and students' profile.
- Enrollment information matrix A combined with pass/completed information without students' metadata U for recommending enrollment based on successful completion of courses.
- Enrollment information matrix A combined with pass/completed information with students' metadata matrix U for recommending enrollment based on successful completion and students' profile.

Experimental results

The experimental results were performed using a training and validating dataset with the enrollment of two disjoint sets of students. Additionally, we split the validating dataset into a set of returning students who had already enrolled in the past, and new students, focusing on predicting the first course to enrol. Since it is a recommender system, we predicted different numbers of recommendations ranging from 1 to 5, and Precision@K, Recall@K and AUC metrics were used for evaluation.

The experimental results assuming only enrollment information without students' metadata for $k = 1, 3$, and 5 are summarized in Table 8.

Number of Recommendations	Returning Students			New Students		
	Precision	Recall	AUC	Precision	Recall	AUC
@K = 1	99%	36%	0.99	17%	17%	0.89
@K = 3	83%	85%	0.99	7%	21%	0.89
@K = 5	59%	94%	0.99	7%	36%	0.89

Table 8. Next course enrollment recommendation without students' metadata.

The experimental results assuming enrollment information with students' metadata for $k = 1, 3$, and 5 are summarized in Table 9.

Number of Recommendations	Returning Students			New Students		
	Precision	Recall	AUC	Precision	Recall	AUC
@K = 1	93%	34%	0.99	17%	17%	0.84
@K = 3	76%	78%	0.99	7%	20%	0.84
@K = 5	57%	90%	0.99	5%	24%	0.84

Table 9. Next course enrollment recommendation with students' metadata.

The experimental results assuming passed/completed information without students' metadata for $k = 1, 3$, and 5 are summarized in Table 10.

Number of Recommendations	Returning Students			New Students		
	Precision	Recall	AUC	Precision	Recall	AUC
@K = 1	96%	21%	0.99	17%	17%	0.85
@K = 3	81%	84%	0.99	7%	21%	0.85
@K = 5	59%	93%	0.99	6%	31%	0.85

Table 10. Next course enrollment recommendation without students' metadata in passed/completed courses.

The experimental results assuming passed/ completed information with students' metadata for $k = 1, 3$, and 5 are summarized in Table 11.

Number of Recommendations	Returning Students			New Students		
	Precision	Recall	AUC	Precision	Recall	AUC
@K = 1	91%	33%	0.99	17%	17%	0.85
@K = 3	76%	78%	0.99	7%	22%	0.85
@K = 5	57%	90%	0.99	5%	24%	0.85

Table 11. Next course enrollment recommendation with students' metadata in passed/completed courses.

The analysis of the results concluded:

- The recommender without metadata information performs better with $k = 3$ on returning students.
- The student's profile providing the number of enrolled courses does not provide enough information to find similar students. Thus, the models complemented with students' metadata perform worse.
- No model is accurate for new students because models cannot find similar students' profiles (with or without characteristics), and they cannot find similarities between courses (because there are no available properties in the model).
- Models with only pass/complete information perform worse because recommendations are based on this information, but students in the validation dataset enrolled without considering this information.
- Adding course metadata may increase the relevancy of the results because students will receive recommendations based on course similarity.

5. Slasys system

5.1. Design a container-based architecture

The infrastructure design is crucial for an effective implementation. Although monolithic infrastructures are still used and have some advantages, infrastructures based on microservices are recognized as a growing good practice because of their major benefits. The system is transformed into packages of small services, each one independently deployable, horizontally scalable, and running its own processes. Moreover, the benefits can be seen during development because there is flexibility to use different technologies and a reduction in development cycles and time-to-market. However, some challenges must be faced because system complexity increases. Microservices need to communicate, and lightweight and secure channels must be provided. Moreover, testing each microservices in an isolated way is more complex, and fault tolerance must be enforced while interacting with other microservices.

Currently, many technology companies are following this paradigm to develop their products. The developed infrastructure follows the microservices infrastructure style supported by DevOps practices:

- **Microservices:** Each service is developed independently and scoped on a single purpose. Docker technology (Anderson, 2015) will containerize each service. Note that Docker allows the deployment of containers in a unique server or on a cloud infrastructure, emulating the microservices and networks.
- **Continuous integration and delivery:** The code is managed with GitLab, and containers will be created automatically within the GitLab server and stored in a private Docker Registry to allow continuous delivery.
- **Infrastructure as a code:** Docker Compose manages the complete infrastructure with a single configuration file, obtaining the latest version of each service from the Docker Registry.
- **Monitoring and Logging:** Docker Compose provides a simple logging process to monitor all the services.

An on-premise server is used to deploy the infrastructure due to the research nature of the project. Nevertheless, the system can be easily reproducible in any cloud environment that supports Docker, Docker Compose, and GitLab technologies.

Multi-tier architecture is used to design the SLASys system. Such architecture is a well-known model used to develop client/server applications. In the case of SLASys, we used different tiers to enclose related microservices. Additionally, it adds security access to some services and stored data by hiding public visibility.

One architecture model was developed for each academy to design custom architectures to meet the academies' needs. The next sections describe the technological developments performed for each academy to gather all the AI-developed components in a unique architecture. The description of all AI components and the different architectures have been summarized in Deliverable 7.

5.2. Talent Academy

The infrastructure design of the system is shown in Figure 18.

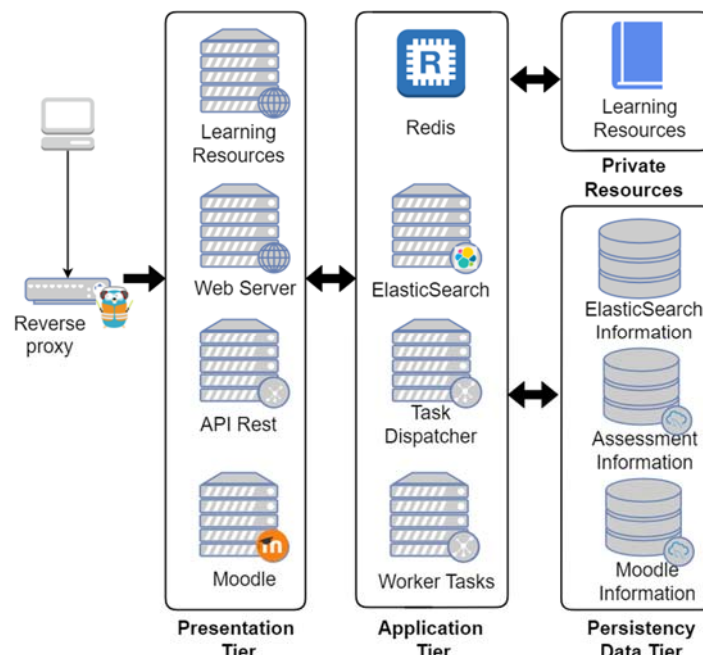


Figure 18. Container-based architecture for Talent Academy.

The components previously described are containerized in a multi-tier architecture. In the presentation tier, multiple accessible services are presented:

- The web server (Django) was used to perform experimental developments in information retrieval (i.e., search on learning resources), answer resources-related questions, and short-answer correctness recommendations.
- API Rest (Python) is used to query and write data related to the different functionalities.
- Moodle (PHP) is used to perform the pilots in short-answer correctness recommendations and information retrieval on SCORM packages.
- Learning resources (Vue.js) is used to access the Course A learning resources securely and allow to perform semantic and literal queries.

All these containers communicate with the application tier, which offers the different functionalities of the system.

- ElasticSearch (Java) supports the semantic search by using LLMs.
- Task Dispatcher (Celery) processes requests performed via API or Web Server and sends the request to the Worker to be processed.
- Worker Tasks (Python) performs the different operations that the infrastructure allows.
- Redis Cache supports the celery communication with the Worker.

The learning resources and database are in the system's private tiers. Learning resources are only accessible from the learning resources interface; meanwhile, data is secured in the persistency data tier and accessible only from other tiers.

5.2.1. API to incorporate the search engine on web-based systems

The search engine developed in Section 3.4.4 was finally developed as a standalone container and included in the REST API with different endpoints (see Figure 19). The API is secured by token authentication and provides different operations to configure the system.

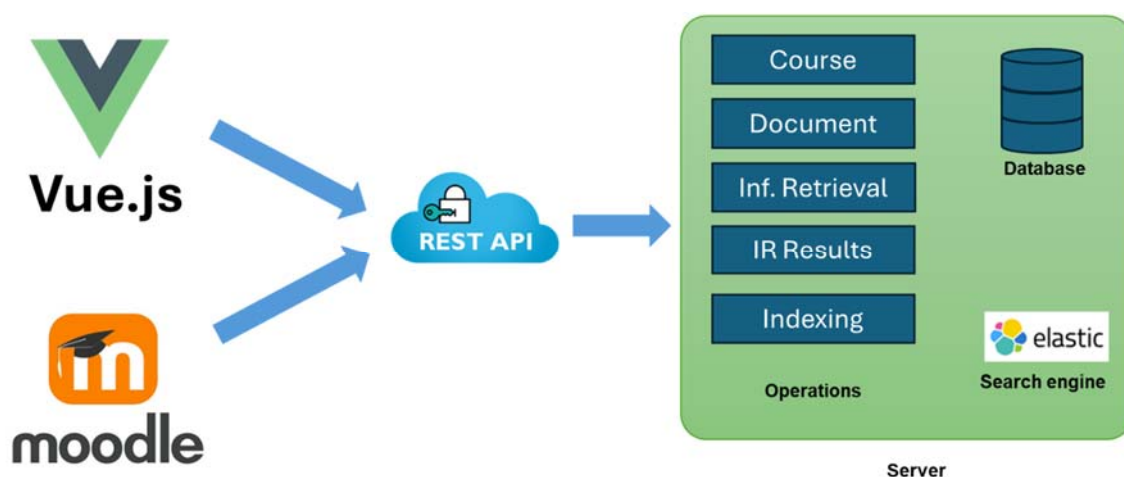


Figure 19. Semantic search REST API.

- Course: An index can be created for each course.
- Indexing: This operation allows to add a new document to an index for a course.
- Document: This operation returns an HTML document from the persistence data tier
- Inf. Retrieval (IR): This operation performs a query request to the ElasticSearch container. It is an asynchronous operation, so it does not wait for the result to avoid an active wait.
- IR Results: When the previous operation is finished, the result is stored in the database. This operation retrieves the result by giving the query's ID.

5.2.2. Search engine integration example 1 (web-based)

As an example of an integration of the search engine, a web application has been implemented in Vue.js (<https://vuejs.org/>). This full Javascript application interacts with the REST API and allows to visualize the learning resources of Course A, and perform searches over those documents. As can be seen in Figure 20, the application is divided into two main parts: the index of contents (1) and the main body (2), where the selected resource is shown. Using the REST API Document endpoint, a secure bridge has been implemented to avoid public access to the learning resources.

Finally, a search button (3) has been included to access the search option, which can be configured by different parameters (see Figure 21). We can first select the maximum number of results per search modality (literal or semantic) (4). Each search modality can be turned on or off, and specific options per modality can be provided:

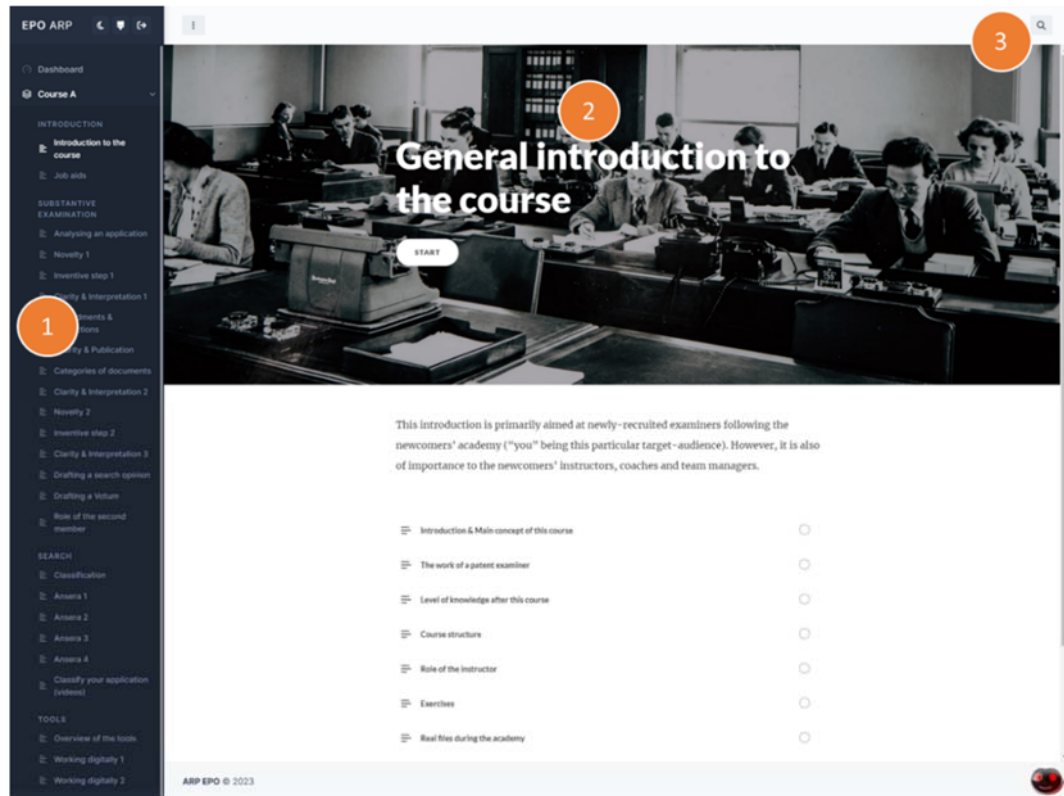


Figure 20. Main page for the web-based search engine.

- In the case of literal search (5), we can select the logic operation between keywords (AND meaning that all keywords must be used or OR if just some of them are required).
- In the case of semantic search (6), we can select the number of candidates that will be considered (this is an optimization option to avoid computing all the distances). Final scores are computed by KNN, where the K value can also be specified.

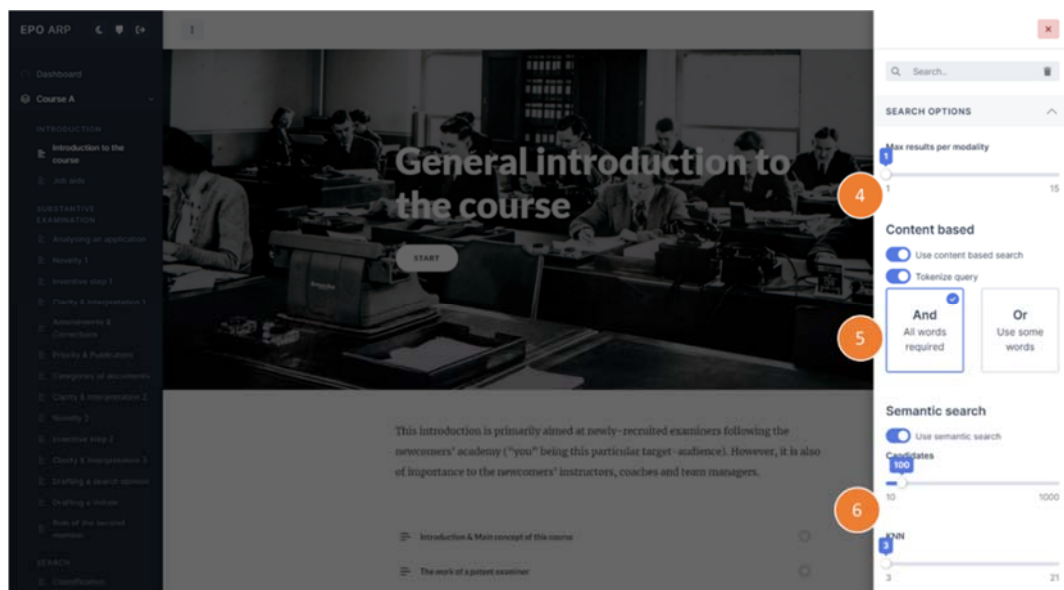


Figure 21. Search options.

The user can provide a search query using the textbox (7) shown in Figure 22. This query and the selected options are sent to the API (i.e., Inf. Retrieval endpoint). The results are obtained by periodically asking the IR Result endpoint until ready, which answers with the list of results (8) when ready and a default score provided by ElasticSearch (i.e., a larger value means a better match of the material with the query). The web interface allows visualizing the specific resource related to each result by clicking on the resulting card.

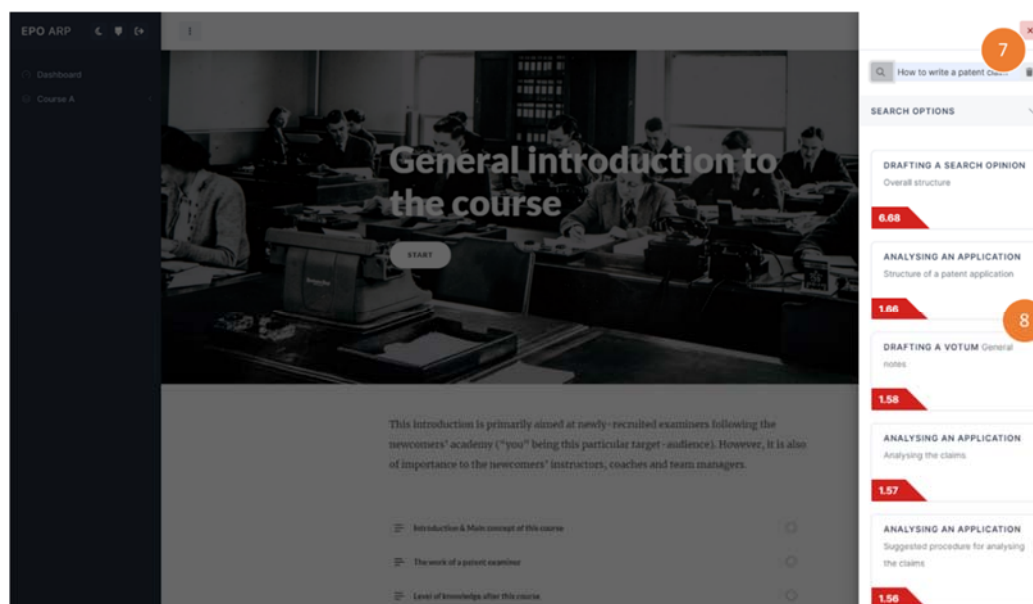


Figure 22. Search query (7) and results (8).

5.2.3. Search engine integration example 2 (Moodle)

The second integration was performed within Moodle, where a search plugin for SCORM packages was developed. Aforementioned, RISE learning resources can also be exported as SCORM packages. SCORM packages can be added to Moodle LMS as learning resources within the scope of a course. However, it does not offer the functionality to search within the complete set of imported SCORM packages.

The developed plugin uses the REST API in the same way as the web-based approach. Figure 23 shows the interface. The search engine has been integrated as a right Moodle block (1), where the index of contents uses the Moodle index (2), and SCORM packages are visualized with the specific module for SCORM packages (3).

The search interface has been designed with the same functionalities as the web-based one (see Figure 24, where functionalities are annotated with the same number as in the web-based one). Semantic and literal searches are configured in the same way. The difference between developments is that Moodle plugin used the indexing endpoint since a course can dynamically add SCORM packages. In the web-based approach, the index was precomputed before its utilization.

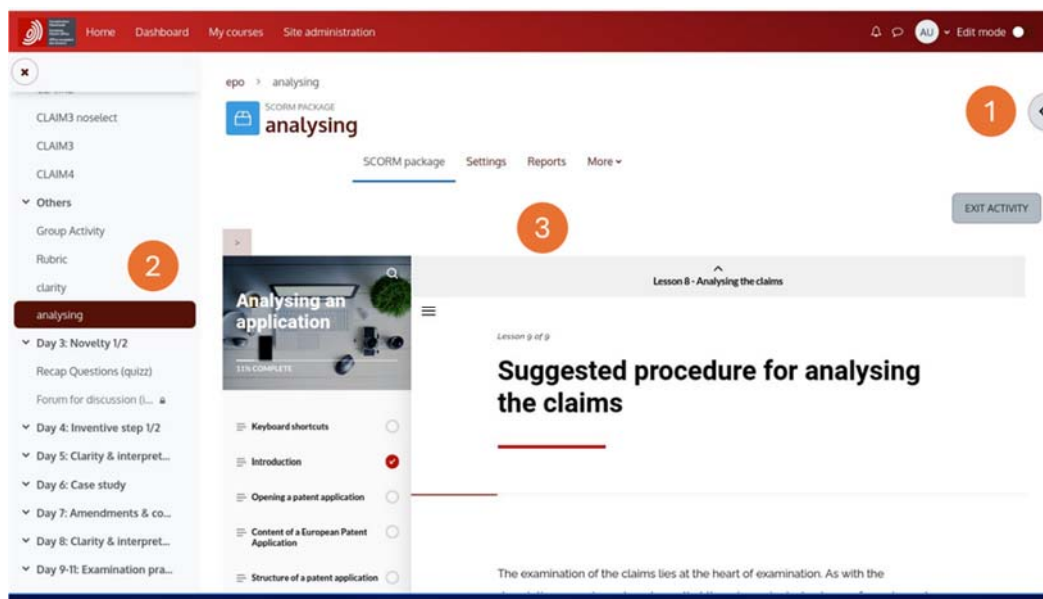


Figure 23. Main page of search engine for Moodle.

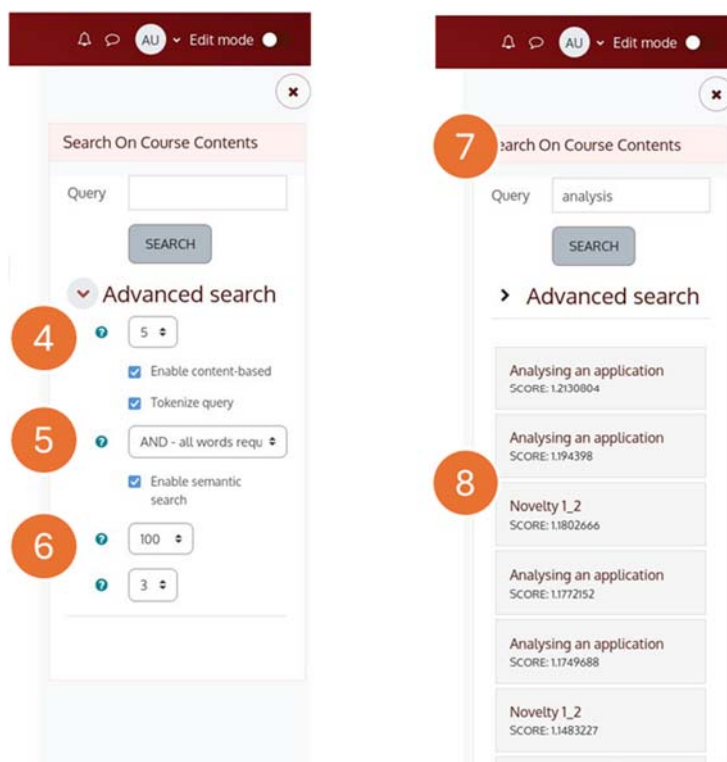


Figure 24. Main page of search engine for Moodle.

5.4.4. API to assess short-answer assessment

The short-answer question recommender has also been integrated into the REST API service with different endpoints (see Figure 25).

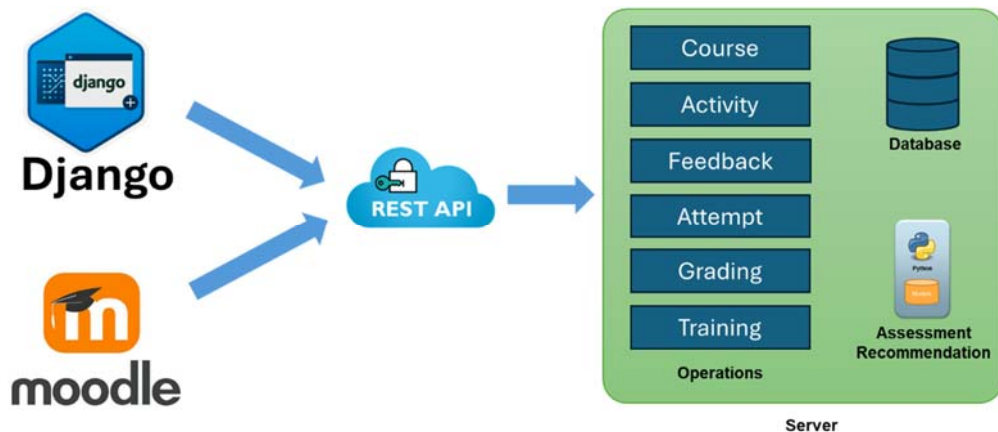


Figure 25. Short-answer assessment recommender REST API.

This API allows the assessment service to be called by any application that supports such technology and uses the standard REST protocol. The API is also secured and provides different operations to configure the system. The allowed operations are:

- **Course:** Define a course where the activities will be defined.
- **Activity:** Set up the activities for the course.
- **Feedback:** Set up the activity configuration, such as the correct answer and the configuration options described in Section 3.5.2.
- **Attempt:** Store the different attempts of the students. The attempt stores the student's answer, the recommended assessment, and the feedback from the assessment system.
- **Grading:** Store the final grading performed by the teacher. This information is used to train the binary classification model since it stores the correct assessment for each answer.
- **Training:** Run the training operation without the intervention of any technical person from the annotated correct and incorrect answers stored in Grading.

5.4.5. Assessment recommender example 1 (web-based)

Similar to the search engine, the REST API provides an interface to use the recommender. Thus, the project provided two integration examples. The first one is a custom web-based application.

A Django web server has been used to design three interfaces: (1) to set up the activity, (2) to send the answer, and (3) to review the recommendation. Figure 26 shows the interface to send an answer to the system. The cog button allows to select the used metric (i.e., semantic comparison or binary classification).

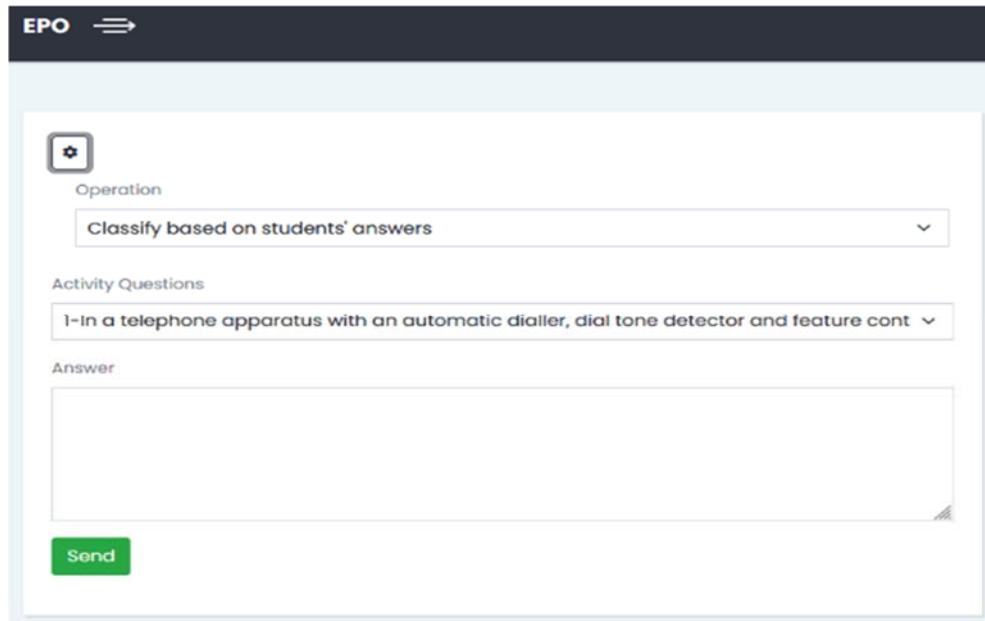
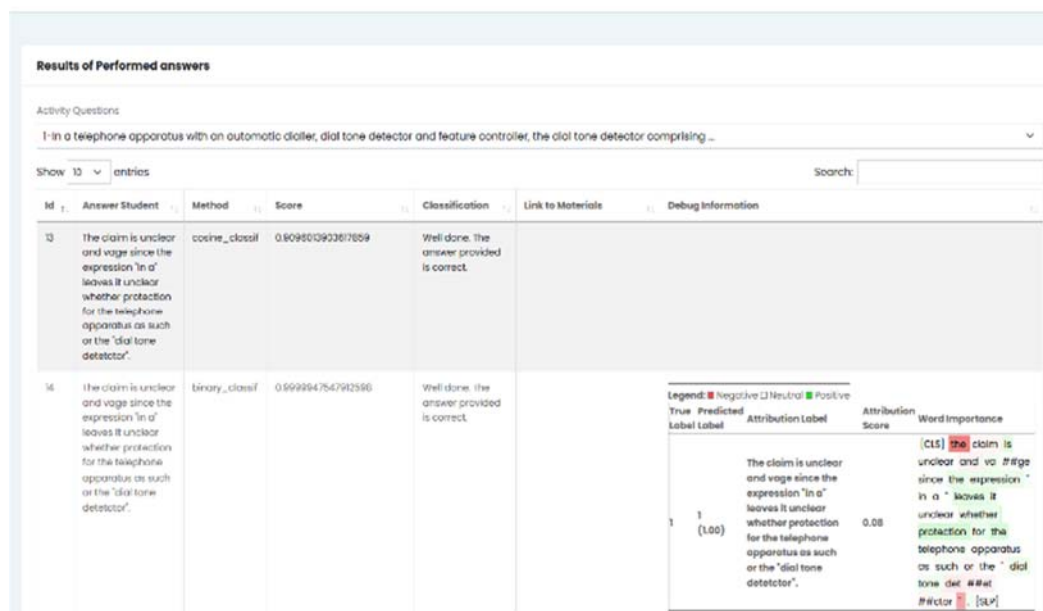


Figure 26. Question answer interface.

Since sending an attempt is an asynchronous operation, the operation does not return the assessment recommendation. The recommendation can be reviewed on the interface shown in Figure 27, which calls to the Attempt endpoint to read the recommendation. The figure illustrates the recommendation result for the two metrics on the same answer. The explainable information provided by the service is shown in the case of the binary classification model.



Id	Answer Student	Method	Score	Classification	Link to Materials	Debug Information
13	The claim is unclear and vague since the expression "in a" leaves it unclear whether protection for the telephone apparatus as such or the "dial tone detector".	cosine_classif	0.80980090367659	Well done, the answer provided is correct.		
14	The claim is unclear and vague since the expression "in a" leaves it unclear whether protection for the telephone apparatus as such or the "dial tone detector".	binary_classif	0.9999947547902596	Well done, the answer provided is correct.		<p>Legend: ■ Negative ■ Neutral ■ Positive</p> <p>True Predicted Attribution Label Label Score Word Importance</p> <p>1 1 (1.00) 0.08 [CLS] the claim is unclear and va #fig since the expression " in a " leaves it unclear whether protection for the telephone apparatus as such or the " dial tone det ##et #factor [LXP]</p>

Figure 27. Answer recommendation revision interface.

5.4.6. Assessment recommender example 2 (Moodle)

The previous interface allows simple integration with the assessment recommender service to practice with the questions. However, it lacks the tracking and grading functionality in a real learning setting. In order to perform a real test with students, the service has been integrated into Moodle.

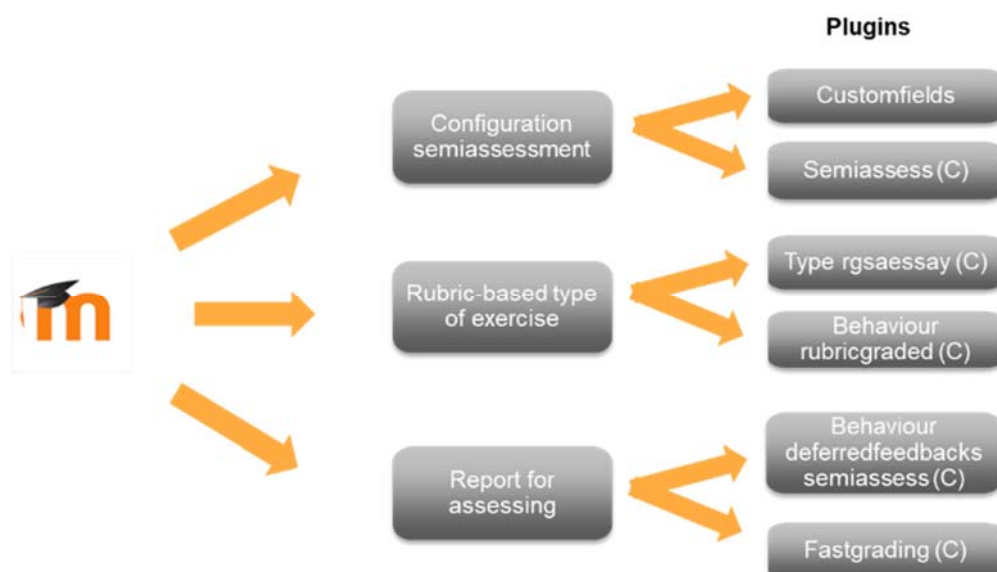


Figure 28. List of implemented plugins on Moodle to support the recommender service.

Since Moodle is highly configurable and allows further development to enhance the LMS, different ad-hoc plugins have been developed (see Figure 28).

- *Customfields* is available on the Moodle plugins portal to support custom fields on activities. Such a plugin is used to specify within Moodle that a quiz has questions that an external assessment service should process.
- *Rgsaessay* (custom) question type is a mixing type between quiz and assignment questions in Moodle. It allows to configure all the information needed to register a new question on the assessment service and incorporate rubric assessment in the quiz questions. Rubric assessment is only available on assignment questions in Moodle. This improvement allows to manually grade questions using a rubric instead of assigning a straightforward grade.
- *Rubricgraded* (custom) behaviour renders the teacher and student view for the rgsaessay question type. It visualizes the assessment recommender response and allows the question grading with the defined rubric.
- *Deferredfeedback* behaviour (custom) configures some feedback parameters to be shown when the teacher's manual grading is done.
- *Fastgrading* (custom) simplifies the grade reporting for rgsaessay questions to show only the latest ungraded students' attempts.
- *semiassess* (custom) plugin communicates with the assessment recommender through the REST API. It supports all the operations to manipulate courses, activities, attempts, and grading. It also processes the attempts and retrieves information from the recommender when the prediction is available.

Figures 29, 30, and 31 show different screenshots of the system. The interface for configuring a question is not shown since it is similar to configuring a Moodle question for quizzes with additional options for adding the annotated answers and providing the feedback information.

- Figure 29 illustrates the student's attempt interface. The question is composed of a choice question and a short-answer field to justify the choice selection. The rgsaessay question supports adding the choice question or only showing the short-answer field to support simple short-answer questions. The figure also shows the rubric to inform the student of the grading criteria.

Question 1

Not yet answered

Marked out of 10.00

Flag question

In a telephone apparatus with an automatic dialler, a dial tone detector and a feature controller, the dial tone detector comprising ...

Select answer

☒ Compliant with the requirements of Art. 84 EPC

☐ Not compliant with the requirements of Art. 84 EPC, because:

Justification answer

↓

A

B

I

Figure 29. Question attempt interface.

- Figure 30 illustrates the teacher grading interface, where we can see how the information of the recommender service has been integrated. The teacher sees the correct answer, the assessment recommendation, the explainable information, the proposed rubric grading (automatically selected depending on the recommendation and recommendation score), the feedback, and the recommendation score (i.e., from 0 to 1). The mark and the feedback is assigned depending on the rubric selection. The teacher must only review this information and select the correct rubric grade.

Model answer

The expression "in" makes it unclear whether protection is sought for the dial tone detector per se or a telephone apparatus (see Guidelines F-IV 4.15).

Correct answer

Assessment Recommendation

Well done. The provided answer is correct.

Assessment recommendation

Legend: ■ Negative □ Neutral ■ Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
1	1 (1.00)	The claim does not clearly defines the subject-matter of the invention, as the term "in" makes it unclear whether the entire telephone apparatus is included in the scope of protection.	-0.25	[CLS] the claim does not clearly defines the subject-matter of the invention, as the term "in" makes it unclear whether the entire telephone apparatus is included in the scope of protection. [SEP]

Explainable information

Rubric (grading)

Response			
The option answer is incorrect	<input type="radio"/>	0 points	
The option answer is correct, but the justification answer is incorrect	<input type="radio"/>	0 points	
The option and justification answers are correct	<input checked="" type="radio"/>	10 points	

Rubric

Comment

Well done. The provided answer is correct.

The correct solution is The expression "in" makes it unclear whether protection is sought for the dial tone detector per se or a telephone apparatus (see Guidelines F-IV 4.15).

Feedback recommendation

Mark

10.00 out of 10.00

Score

1.000

Figure 30. Answer grading interface.

- Figure 31 shows the student's review interface after the question is graded. The student sees the mark and the feedback based on the selected rubric option validated by the teacher. Additionally, the student can see the assessment of the choice question when it is available.

In a telephone apparatus with an automatic dialler, a dial tone detector and a feature controller, the dial tone detector comprising ...

Select answer

☐ Compliant with the requirements of Art. 84 EPC ✖

☒ Not compliant with the requirements of Art. 84 EPC, because: ✔

Assessment option answer

Justification answer

The claim does not clearly defines the subject-matter of the invention, as the term "in" makes it unclear whether the entire telephone apparatus is included in the scope of protection.

Comment: Well done. The provided answer is correct.

Feedback

The correct solution is *The expression "in" makes it unclear whether protection is sought for the dial tone detector per se or a telephone apparatus (see Guidelines F-IV 4.15).*

Rubric (review)

Rubric assessment

Response	The option answer is incorrect	The option answer is correct, but the justification answer is incorrect	The option and justification answers are correct
	0 points	0 points	10 points

Mark

Figure 31. Student's answer review interface.

5.3. Patent Academy

The infrastructure of the Slasys Project for Patent Academy is summarized in Figure 32 with the following microservices:

- The *Web Service* displays all the information gathered within the system. Predictive and analytical information is combined in the different dashboards. It is the only public service.
- The *R service* performs statistical analysis and computes the models' statistical performance.
- The *Restful API* provides access to all operations forwarded to the data-independent tier, such as serving data to the web service, configuring the system for the institution, and running operations related to the predictive models.
- Redis* is used to cache some query operations and improve efficiency.
- The *Computational* service performs all training, testing and predictive operations on LMS data.
- The *Loader* service exports data from the LMS and transforms it into a data-independent format to be processed by the *Computational* service
- The *Anonymizer* provides access to the anonymizer function to obtain the hash identifier used to store the data related to each student within the MongoDB.
- MongoDB and MySQL databases are used to store all persistent data.

- The *Mail* and *Recommender* Service are out of the development scope since Slasys is set in simulation mode.

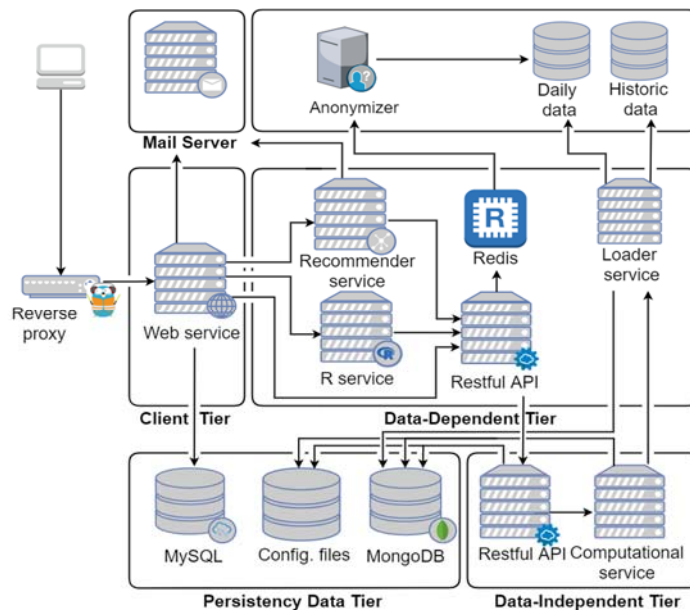


Figure 32. Container-based architecture for Patent Academy.

The web interface allows the configuration of the Slasys system and shows training accuracy, at-risk identification and performance on such identification. The next sections highlights the more relevant functionalities.

5.3.1 Trained models accuracy

Accuracy metrics can be reviewed for all trained models. Figure 33 illustrates an example of the failure models. These reports show relevant information about the trained models and their accuracy metrics.

Download New Bayes Download Decision Tree Download Support Vector Machine Download K-nearest neighbor Download Intrasemester Download Intersemester Download optimum Download New BayesError Download Decision TreeError Download Support Vector MachineError Download K-nearest neighbourError																			
Show 25 entries																			
List of models trained correctly																			
Subject	Name	Activity	Model	Features	Training test.	Test test.	Acc	MAE	RMSE	TH	FN	TP	FP	THR	TPR	Fscore1.5	AUC	Algorithm	Semester
28	Practical and strategical aspects of the CPC	1	STUDENT_PASS_COURSE_ACT1	5	214	214	94.86	0.05	0.23	2	10	201	1	66.67	95.26	96.53	0.81	Naive Bayes	3
28	Practical and strategical aspects of the CPC	2	STUDENT_PASS_COURSE_ACT2	6	214	214	84.58	0.15	0.39	3	33	178	0	100	84.36	88.63	0.92	Naive Bayes	3
30	Patentability requirements at the EPO	1	STUDENT_PASS_COURSE_ACT1	5	757	759	98.68	0.01	0.11	15	6	734	4	78.95	99.19	99.27	0.89	Naive Bayes	3
31	Using CPC in classification	1	STUDENT_PASS_COURSE_ACT1	5	572	572	99.48	0.01	0.07	0	0	569	3	0	100	99.84	0.5	Decision Tree	3
31	Using CPC in classification	2	STUDENT_PASS_COURSE_ACT2	6	572	572	99.48	0.01	0.07	0	0	569	3	0	100	99.84	0.5	K-Nearest Neighbors	3
31	Using CPC in classification	3	STUDENT_PASS_COURSE_ACT3	7	572	572	99.48	0.01	0.07	0	0	569	3	0	100	99.84	0.5	K-Nearest Neighbors	3
31	Using CPC in classification	4	STUDENT_PASS_COURSE_ACT4	8	572	572	78.67	0.21	0.46	1	120	449	2	33.33	78.91	84.29	0.56	Naive Bayes	3

Figure 33. Individual course training accuracy for failure models.

The next activity recommendation and dropout models also share the same interface. However, particular information was also provided for the dropout model about the accuracy detection across the course timeline. Figure 34 illustrates the plots for accuracy, TPR and F-score 1.5 metrics across the course timeline ordered from the farthest accessed day (i.e., the first access to the course) until the day the students stop accessing the course. Data is aggregated. Thus, some students stayed 51 days in the course, while others stayed fewer days. We can observe that the accuracy increases when fewer days remain to stop

accessing the course. The model's accuracy is approximately 80% the day the students stop accessing the course (i.e., day 0).

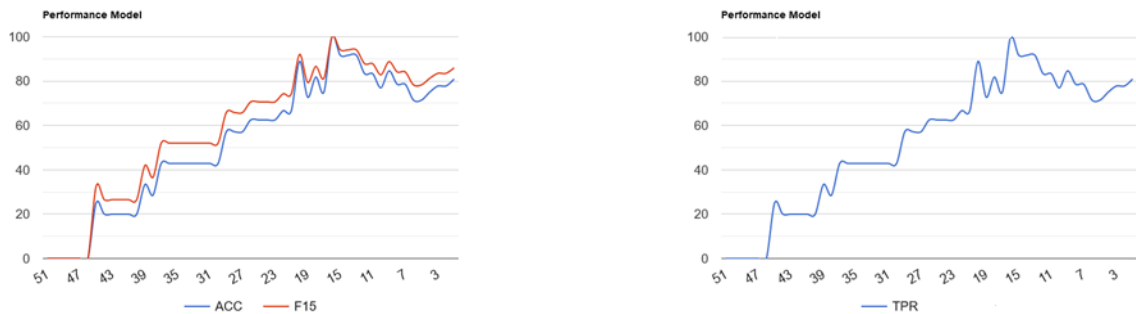


Figure 34. Individual course training accuracy for dropout model.

5.3.2. Analytical view

The analytical view offers the capability to analyze the students' performance and the at-risk identification. The analysis could be done using two views:

- Dashboard: Aggregated analysis for a course.
- At-risk individual identification: Individual analysis for each student for a course.

Dashboard

The dashboard shows aggregated information for the failure and dropout model combined with analytical information about the performance and Moodle utilization. Figure 35 depicts the different elements of the dashboard:

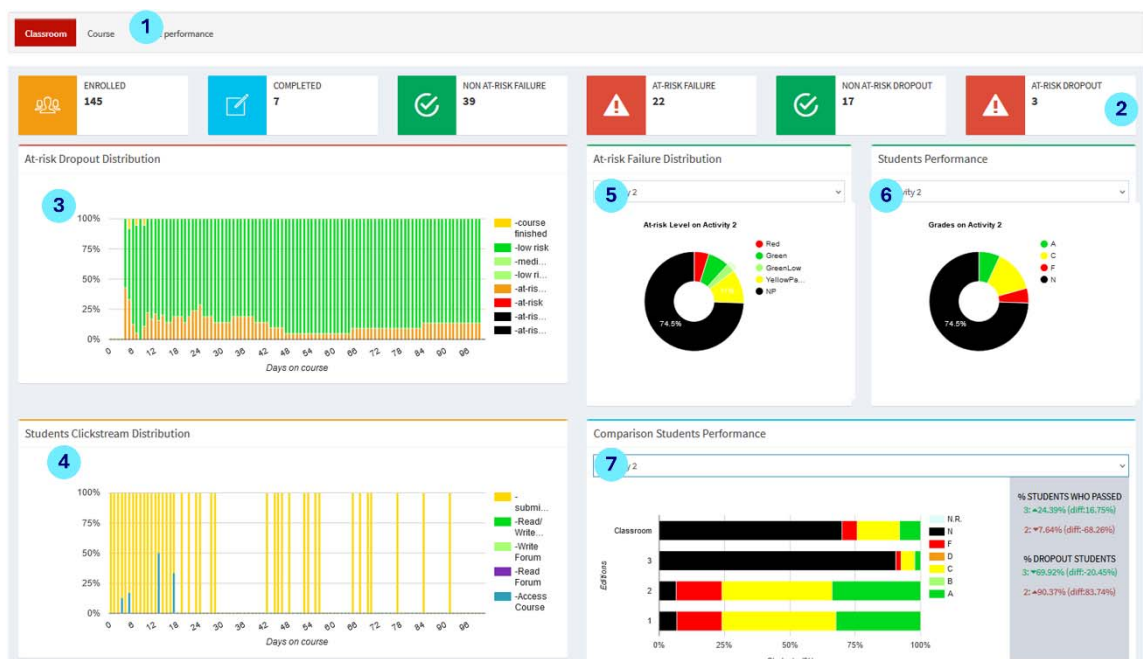


Figure 35. Aggregated dashboard for a course.

1. The information can be aggregated by classroom or by course.
2. Information about the number of enrolled students, students who completed the course (if the information exists), and the number of non-at-risk and at-risk students for both models is shown.
3. Daily evolution of the dropout at-risk within the course. Instead of using dates, the plot aggregates by the number of days from the enrollment day, assuming as 1 the first day a student accesses the course. Since many courses are open and students enrol at any time, this visualization better describes the evolution of the at-risk depending on the number of days a student is within the course.
4. Daily analytical information about the student's actions within the course. This plot shows which actions students perform and when they do them. The collected events are access to the course, activity submission, and access to the forums. Complete and finish events are shown in the daily evolution of the dropout at-risk.
5. Distribution of the failure prediction by activity. This distribution also shows the inactive students who did not submit the selected activity.
6. Distribution of the activity grades/submissions. Similar to the previous plot, inactive students are also counted.
7. Comparison of the grade/submission distribution between editions of the same Moodle course. When selecting a classroom, the grade/submission distribution of the classroom is also compared with the total course performance.

Individual at-risk identification

Teachers also have at-risk levels detailed information for each student at the classroom level. The report uses a tabular representation to report the at-risk levels. First, the profile information is shown (see Figure 36):

Students of the classroom			
User	Name	Last login	Date Enrollment / Completed
 AlSmith1	Alyssa Smith Enrolled: 1 Repeated: 0 Completed: 0	10/02/2023	07/02/2023
 AmBrown5	Amanda Brown Enrolled: 1 Repeated: 0 Completed: 0	02/02/2023	27/01/2023 / 02/02/2023
 AsMadden	Ashley Madden Enrolled: 2 Repeated: 0 Completed: 1	10/02/2023	10/02/2023

Figure 36. Profile information in the individual at-risk report.

- Profile information: The profile information used to compute the predictions is shown. The information gathered from Moodle is how many courses the student has enrolled in the past, the number of repeated times the current course, and how many passed and completed courses.
- The last login date in Moodle.
- Enrollment and completed date when available.

Related to the failure model, the dashboard illustrates the activities performed by the student in sequential order (see Figure 37). Each activity already submitted (and assessed with a grade when the information is available) is shown with additional information. Activities not yet submitted are shown in a bubble without any at-risk level colour.

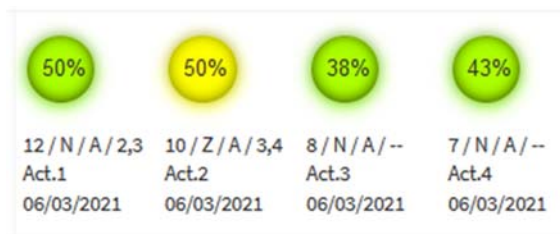


Figure 37. At-risk failure predictions for a student.

- The at-risk level color is assigned based on the failure at-risk identification, explained in Section 4.4.2.
- The percentage of students with the same profile and at-risk level who finally passed/completed the course in the past is shown within the at-risk level colour.
- Below the at-risk colour, there is information about the number of students in the training dataset with the same profile and at-risk identification. This information gives insights about whether this at-risk has been previously identified in the training dataset or a new type is appearing.
- Additionally, the grade predicted as the minimum grade to pass the course and the grade obtained by the student are also displayed. For models based on the submission event, it shows whether submitting the activity impacts the pass/complete event (i.e., whether the predictive model considers the activity mandatory to pass the course).
- Finally, the recommended next activity (or list of activities) is shown.

Finally, the dropout at-risk and the analytical information about the student's actions are shown in two corresponding bars (see Figure 38).



Figure 38. At-risk dropout predictions for a student.

- At-risk dropout identification is shown in the first bar. There is a prediction for each day, and a blue dash is shown when there is insufficient information to perform a prediction. Non-at-risk predictions are depicted above the bar, and at-risk predictions are below the bar when the interval confidence is starting to be consumed (i.e., light green colour) or has already been consumed (i.e., red colour). The colours are the ones described in Section 4.5.2. The bar also depicts when the student has completed or passed the course with a golden colour pin. The prediction starts when the dropout model has enough information to compute predictions. Usually, the predictions start when there are two events within a minimum of 5 days (i.e., a requirement to run the WTTE model).

- Analytical information is shown in the second bar. This bar depicts the events captured by the model. Access information, submission dates with the corresponding activities, and forum access are shown on the corresponding days.

5.3.3. At-risk identification performance

This functionality summarizes the at-risk detection performance for the simulated students of the testing dataset for an individual course. The report offers the following information:

Results of the failing predictive model

Showing 1 to 3 of 3 entries

Show 50 entries

Search:

Activity	TN	FN	TP	FP	ACC	TPR	TNR	F1S
Activity_1	0	0	425	910	31.84	100	0	60.28
Activity_2	182	1	424	728	45.39	99.76	20	65.36
Activity_3	909	2	423	1	99.78	99.53	99.89	99.6

Showing 1 to 3 of 3 entries

Show 50 entries

Search:

Activity	Metric	Green	GreenLow	YPassLow	YPassActivity	YNotPassLow	Red	NotSubmitted	NotSubmitted=2
Message_1	Total	0	0	0	0	0	0	1335	0
Message_1	Completed	0	0	0	0	0	0	910	0
Message_1	% Completed	0	0	0	0	0	0	68.16	0
Message_2	Total	169	14	0	2	0	10	1140	0
Message_2	Completed	168	14	0	1	0	0	727	0
Message_2	% Completed	99.42	100	0	50	0	0	63.77	0
Message_3	Total	848	63	0	2	0	64	358	0
Message_3	Completed	846	63	0	1	0	0	0	0
Message_3	% Completed	99.76	100	0	50	0	0	0	0

Showing 1 to 9 of 9 entries

Previous 1 Next

Figure 39. At-risk failure identification performance.

- Failure model regarding completeness: The report is split into two tables (see Figure 39), which summarize (1) the prediction accuracy regarding the pass/complete event (i.e., to check whenever the model outcome fail/pass successfully detects student final failure/completeness); and (2) the distribution of students' completeness regarding the different at-risk levels (i.e., students who complete the course should be mainly assigned to low at-risk levels).
- Dropout model regarding completeness: This report is similar to the previous ones but checks whenever the at-risk dropout alarm is not raised on students who passed/ completed the course.
- Dropout model regarding future access to the course: The previous model gives insights on courses with ending events. This report complements the course performance evaluation on courses with no ending event by checking whether students access the course after the at-risk dropout alarm. Thus, this report assumes model success when the at-risk dropout is triggered and the student does not access the course again.

6. Conclusions

This document describes the different outputs of the SLASys research project. The outputs have been summarized, and the obtained insights have been provided. All outputs aim to solve the challenges stated in the initial project proposal and concerns raised by the EPO team during the project.

The project has proposed a semantic search engine and explored different techniques. Although technology and research related to LLM are quickly advancing, this project has proved how it can be integrated for private use for organizations without requiring the utilization of paid services and in servers with limited resources. The proposed approach has been integrated into a REST API for further use in other services, and two integration examples have been proposed.

The short-answer assessment recommender has been designed with the same principles: free models for private use, run in servers with limited resources, and without technical people's intervention to train new models. The recommender is also provided with a REST API, and two integration examples have also been proposed. The most remarkable is the integration with Moodle, which allows tracking users and grading the knowledge they acquire.

At-risk identification has been the third main outcome of the project. A complete predictive analytics infrastructure has been developed. Such infrastructure can help teachers better understand students' problems based on their interaction with the Moodle LMS and be proactive in solving them.

All outcomes have been tested with data of real students enforcing their utilization in real settings. High accuracy has been obtained in some evaluations (i.e., short-answer correctness recommendation and detection of at-risk failure models). Other approaches require further exploration and fine-tuning of the gathered data to improve accuracy, which are set as future work.

Bibliography

- Adhikari, A., Ram, A., Tang, R., & Lin, J. (2019). *DocBERT: BERT for Document Classification*.
- Alamri, A., Sun, Z., Cristea, A. I., Stewart, C., & Pereira, F. D. (2021). MOOC Next Week Dropout Prediction: Weekly Assessing Time and Learning Patterns. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12677 LNCS, 119–130. https://doi.org/10.1007/978-3-030-80421-3_15
- Anderson, C. (2015). Docker. In *IEEE Software* (Vol. 32, Issue 3). <https://doi.org/10.1109/MS.2015.62>
- Arnold, K. E., & Pistilli, M. D. (2012). Course signals at Purdue: Using learning analytics to increase student success. *ACM International Conference Proceeding Series*, 267–270. <https://doi.org/10.1145/2330601.2330666>
- Azcona, D., & Casey, K. (2015). Micro-analytics for student performance prediction leveraging fine-grained learning analytics to predict performance. *International Journal of Computer Science and Software Engineering*, 4(8)(8).
- Bağrıacık Yılmaz, A., & Karataş, S. (2022). Why do open and distance education students drop out? Views from various stakeholders. *International Journal of Educational Technology in Higher Education* 2022 19:1, 19(1), 1–22. <https://doi.org/10.1186/S41239-022-00333-X>
- Bahdanau, D., Cho, K. H., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.
- Baneres, D., Rodríguez, M. E., Guerrero-Roldán, A. E., & Karadeniz, A. (2020). An early warning system to detect at-risk students in online higher education. *Applied Sciences (Switzerland)*, 10(13), 4427. <https://doi.org/10.3390/app10134427>
- Bañeres, D., Rodríguez-González, M. E., Guerrero-Roldán, A. E., & Cortadas, P. (2023). An early warning system to identify and intervene online dropout learners. *International Journal of Educational Technology in Higher Education*, 20(1). <https://doi.org/10.1186/s41239-022-00371-5>
- Beltagy, I., Lo, K., & Cohan, A. (2019). SCIBERT: A pretrained language model for scientific text. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*. <https://doi.org/10.18653/v1/d19-1371>
- Black, S., Leo, G., Wang, P., Leahy, C., & Biderman, S. (2021). *GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow*. Zenodo. <https://doi.org/10.5281/zenodo.5297715>

- Boudjehem, R., & Lafifi, Y. (2021). A new approach to identify dropout learners based on their performance-based behavior. *Journal of Universal Computer Science*, 27(10). <https://doi.org/10.3897/jucs.74280>
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 2020-December.
- Burrows, S., Gurevych, I., & Stein, B. (2015). The eras and trends of automatic short answer grading. In *International Journal of Artificial Intelligence in Education* (Vol. 25, Issue 1). <https://doi.org/10.1007/s40593-014-0026-8>
- Calvo-Flores, M. D., Galindo, E. G., Jiménez, M. C. P., & Pérez, O. (2006). Predicting students' marks from Moodle logs using neural network models. *Proceedings of the IV International Conference on Multimedia and Information and Communication Technologies in Education (M-ICTEE2006)*, 1.
- Candel, A., McKinney, J., Singer, P., Pfeiffer, P., Jeblick, M., Lee, C. M., & Conde, M. V. (2023). H2O Open Ecosystem for State-of-the-art Large Language Models. *EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings of the System Demonstrations*. <https://doi.org/10.18653/v1/2023.emnlp-demo.6>
- Carbonell, J., & Goldstein, J. (1998). The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. *SIGIR 1998 - Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. <https://doi.org/10.1145/290941.291025>
- Casey, K., & Azcona, D. (2017). Utilizing student activity patterns to predict performance. *International Journal of Educational Technology in Higher Education*, 14(1). <https://doi.org/10.1186/s41239-017-0044-3>
- Cerezo, R., Sánchez-Santillán, M., Paule-Ruiz, M. P., & Núñez, J. C. (2016). Students' LMS interaction patterns and their relationship with achievement: A case study in higher education. *Computers and Education*, 96, 42–54. <https://doi.org/10.1016/j.compedu.2016.02.006>
- Chen, Y., Chen, Q., Zhao, M., Boyer, S., Veeramachaneni, K., & Qu, H. (2017). DropoutSeer: Visualizing learning patterns in Massive Open Online Courses for dropout reasoning and prediction. *2016 IEEE Conference on Visual Analytics Science and Technology, VAST 2016 - Proceedings*. <https://doi.org/10.1109/VAST.2016.7883517>
- Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: PRE-TRAINING TEXT ENCODERS AS DISCRIMINATORS RATHER THAN GENERATORS. *8th International Conference on Learning Representations, ICLR 2020*.
- Dalipi, F., Imran, A. S., & Kastrati, Z. (2018). MOOC dropout prediction using machine learning techniques: Review and research challenges. *IEEE Global Engineering*

- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1.
- Dourado, R. A., Rodrigues, R. L., Ferreira, N., Mello, R. F., Gomes, A. S., & Verbert, K. (2021). A teacher-facing learning analytics dashboard for process-oriented feedback in online learning. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3448139.3448187>
- El-Sabagh, H. A. (2021). Adaptive e-learning environment based on learning styles and its impact on development students' engagement. *International Journal of Educational Technology in Higher Education*, 18(1). <https://doi.org/10.1186/s41239-021-00289-4>
- Fei, M., & Yeung, D. Y. (2016). Temporal Models for Predicting Student Dropout in Massive Open Online Courses. *Proceedings - 15th IEEE International Conference on Data Mining Workshop, ICDMW 2015*, 256–263. <https://doi.org/10.1109/ICDMW.2015.174>
- Freitas, R., & Salgado, L. (2020). Educators in the loop: Using scenario simulation as a tool to understand and investigate predictive models of student dropout risk in distance learning. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12217 LNCS, 255–272. https://doi.org/10.1007/978-3-030-50334-5_17
- Gašević, D., Dawson, S., Rogers, T., & Gasevic, D. (2016). Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success. *Internet and Higher Education*, 28. <https://doi.org/10.1016/j.iheduc.2015.10.002>
- Gligorea, I., Cioca, M., Oancea, R., Gorski, A. T., Gorski, H., & Tudorache, P. (2023). Adaptive Learning Using Artificial Intelligence in e-Learning: A Literature Review. In *Education Sciences* (Vol. 13, Issue 12). <https://doi.org/10.3390/educsci13121216>
- Goel, Y., & Goyal, R. (2020). On the Effectiveness of Self-Training in MOOC Dropout Prediction. *Open Computer Science*, 10(1), 246–258. <https://doi.org/10.1515/comp-2020-0153>
- Grau-Valldosera, J., & Minguillón, J. (2014). Rethinking dropout in online higher education: The case of the universitat oberta de catalunya. *International Review of Research in Open and Distance Learning*, 15(1), 290–308. <https://doi.org/10.19173/irrodl.v15i1.1628>
- Greenland, S. J., & Moore, C. (2022). Large qualitative sample and thematic analysis to redefine student dropout and retention strategy in open online education. *British Journal of Educational Technology*, 53(3). <https://doi.org/10.1111/bjet.13173>
- Gueniche, T., Fournier-Viger, P., & Tseng, V. S. (2013). Compact prediction tree: A lossless model for accurate sequence prediction. *Lecture Notes in Computer Science*

(Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8347 LNAI(PART 2). https://doi.org/10.1007/978-3-642-53917-6_16

- Ho, L. L., & Silva, A. F. (2006). Unbiased estimators for mean time to failure and percentiles in a Weibull regression model. *International Journal of Quality and Reliability Management*, 23(3). <https://doi.org/10.1108/02656710610648251>
- Howard, E., Meehan, M., & Parnell, A. (2018). Contrasting prediction methods for early warning systems at undergraduate level. *Internet and Higher Education*, 37. <https://doi.org/10.1016/j.iheduc.2018.02.001>
- Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LORA: LOW-RANK ADAPTATION OF LARGE LANGUAGE MODELS. *ICLR 2022 - 10th International Conference on Learning Representations*.
- Hu, Y. H., Lo, C. L., & Shih, S. P. (2014). Developing early warning systems to predict students' online learning performance. *Computers in Human Behavior*, 36, 469–478. <https://doi.org/10.1016/j.chb.2014.04.002>
- Huang, S., & Fang, N. (2013). Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models. *Computers and Education*, 61(1), 133–145. <https://doi.org/10.1016/j.compedu.2012.08.015>
- Itani, A., Brisson, L., & Garlatti, S. (2018). Understanding Learner's Drop-Out in MOOCs. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11314 LNCS. https://doi.org/10.1007/978-3-030-03493-1_25
- Joksimović, S., Gašević, D., Loughin, T. M., Kovanović, V., & Hatala, M. (2015). Learning at distance: Effects of interaction traces on academic achievement. *Computers and Education*, 87. <https://doi.org/10.1016/j.compedu.2015.07.002>
- Knowles, J. (2014). Of Needles and Haystacks: Building an Accurate Statewide Dropout Early Warning System in Wisconsin. *JEDM - Journal of Educational Data Mining*, 7(3), 1–52. <https://doi.org/10.5281/zenodo.3554725>
- Kotsiantis, S. B., Pierrakeas, C. J., & Pintelas, P. E. (2003). Preventing student dropout in distance learning using machine learning techniques. *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 2774 PART, 267–274. https://doi.org/10.1007/978-3-540-45226-3_37
- Kula, M. (2015). Metadata embeddings for user and item cold-start recommendations. *CEUR Workshop Proceedings*, 1448.
- Kusal, S., Patil, S., Choudrie, J., Kotecha, K., Mishra, S., & Abraham, A. (2022). AI-based Conversational Agents: A Scoping Review from Technologies to Future Directions. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2022.3201144>
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE

REPRESENTATIONS. *8th International Conference on Learning Representations, ICLR 2020.*

- Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4). <https://doi.org/10.1093/bioinformatics/btz682>
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. <https://doi.org/10.18653/v1/2020.acl-main.703>
- López-Zambrano, J., Lara, J. A., & Romero, C. (2020). Towards portability of models for predicting students' final performance in university courses starting from moodle logs. *Applied Sciences (Switzerland)*, 10(1). <https://doi.org/10.3390/app10010354>
- Lykourantzou, I., Giannoukos, I., Nikolopoulos, V., Mpardis, G., & Loumos, V. (2009). Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers and Education*, 53(3), 950–965. <https://doi.org/10.1016/j.compedu.2009.05.010>
- Macfadyen, L. P., & Dawson, S. (2010). Mining LMS data to develop an “early warning system” for educators: A proof of concept. *Computers and Education*, 54(2), 588–599. <https://doi.org/10.1016/j.compedu.2009.09.008>
- Marbouti, F., Diefes-Dux, H. A., & Madhavan, K. (2016). Models for early prediction of at-risk students in a course using standards-based grading. *Computers and Education*, 103, 1–15. <https://doi.org/10.1016/j.compedu.2016.09.005>
- Márquez-Vera, C., Cano, A., Romero, C., Noaman, A. Y. M., Mousa Fardoun, H., & Ventura, S. (2016). Early dropout prediction using data mining: A case study with high school students. *Expert Systems*, 33(1), 107–124. <https://doi.org/10.1111/exsy.12135>
- Mishra, T., Kumar, D., & Gupta, S. (2014). Mining students' data for prediction performance. *International Conference on Advanced Computing and Communication Technologies, ACCT*. <https://doi.org/10.1109/ACCT.2014.105>
- Moreno-Marcos, P. M., Alario-Hoyos, C., Muñoz-Merino, P. J., & Kloos, C. D. (2019). Prediction in MOOCs: A Review and Future Research Directions. *IEEE Transactions on Learning Technologies*, 12(3). <https://doi.org/10.1109/TLT.2018.2856808>
- Mousavinasab, E., Zarifsanaiey, N., R. Niakan Kalhori, S., Rakhshan, M., Keikha, L., & Ghazi Saeedi, M. (2021). Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. In *Interactive Learning Environments* (Vol. 29, Issue 1, pp. 142–163). <https://doi.org/10.1080/10494820.2018.1558257>
- Mubarak, A. A., Cao, H., & Hezam, I. M. (2021). Deep analytic model for student dropout prediction in massive open online courses. *Computers & Electrical Engineering*, 93, 107271. <https://doi.org/10.1016/j.compeleceng.2021.107271>

- Mubarak, A. A., Cao, H., & Zhang, W. (2020). Prediction of students' early dropout based on their interaction logs in online learning environment. *Interactive Learning Environments*. <https://doi.org/10.1080/10494820.2020.1727529>
- Najdi, L., & Er-Raha, B. (2016). A Novel Predictive Modeling System to Analyze Students at Risk of Academic Failure. *International Journal of Computer Applications*, 156(6), 25–30. <https://doi.org/10.5120/ijca2016912482>
- Nguyen, H. A., Bhat, S., Moore, S., Bier, N., & Stamper, J. (2022). Towards Generalized Methods for Automatic Question Generation in Educational Domains. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 13450 LNCS. https://doi.org/10.1007/978-3-031-16290-9_20
- OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., ... Zoph, B. (2023). *GPT-4 Technical Report*.
- Ortigosa, A., Carro, R. M., Bravo-Agapito, J., Lizcano, D., Alcolea, J. J., & Blanco, Ó. (2019). From Lab to Production: Lessons Learnt and Real-Life Challenges of an Early Student-Dropout Prevention System. *IEEE Transactions on Learning Technologies*, 12(2), 264–277. <https://doi.org/10.1109/TLT.2019.2911608>
- Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). (OpenAI Transformer): Improving Language Understanding by Generative Pre-Training. *OpenAI*.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21.
- Rastrollo-Guerrero, J. L., Gómez-Pulido, J. A., & Durán-Domínguez, A. (2020). Analyzing and predicting students' performance by means of machine learning: A review. In *Applied Sciences (Switzerland)* (Vol. 10, Issue 3). <https://doi.org/10.3390/app10031042>
- Romero, C., López, M. I., Luna, J. M., & Ventura, S. (2013). Predicting students' final performance from participation in on-line discussion forums. *Computers and Education*, 68. <https://doi.org/10.1016/j.compedu.2013.06.009>
- Saarela, M., & Ark Ainen, T. (2015). Analysing Student Performance using Sparse Data of Core Bachelor Courses. *Journal of Educational Data Mining*, 7(1).
- Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*.
- Saura, J. R., Reyes-Menendez, A., & Bennett, D. (2019). How to extract meaningful insights from UGC: A knowledge-based method applied to education. *Applied Sciences (Switzerland)*, 9(21). <https://doi.org/10.3390/app9214603>

- Siemens, G., & Baker, R. S. J. D. (2012). Learning analytics and educational data mining: Towards communication and collaboration. *ACM International Conference Proceeding Series*, 252–254. <https://doi.org/10.1145/2330601.2330661>
- Simpson, C., Baker, K., & Mellinger, G. (1980). Conventional Failures and Unconventional Dropouts: Comparing Different Types of University Withdrawals. *Sociology of Education*, 53(4). <https://doi.org/10.2307/2112529>
- Souza, F., Nogueira, R., & Lotufo, R. (2019). *Portuguese Named Entity Recognition using BERT-CRF*.
- Srilekshmi, M., Sindhumol, S., Chatterjee, S., & Bijlani, K. (2017). Learning Analytics to Identify Students At-risk in MOOCs. *Proceedings - IEEE 8th International Conference on Technology for Education, T4E 2016*, 194–199. <https://doi.org/10.1109/T4E.2016.048>
- Stone, C., & O'Shea, S. (2019). Older, online and first: Recommendations for retention and success. *Australasian Journal of Educational Technology*, 35(1), 57–69. <https://doi.org/10.14742/ajet.3913>
- Tang, J. K. T., Xie, H., & Wong, T. L. (2015). A big data framework for early identification of dropout students in MOOC. *Communications in Computer and Information Science*, 559. https://doi.org/10.1007/978-3-662-48978-9_12
- Thalhammer, V., Hoffmann, S., von Hippel, A., & Schmidt-Hertha, B. (2022). Dropout in adult education as a phenomenon of fit—an integrative model proposal for the genesis of dropout in adult education based on dropout experiences. *European Journal for Research on the Education and Learning of Adults*.
- Tinto, V. (1975). Dropout from Higher Education: A Theoretical Synthesis of Recent Research. *Review of Educational Research*, 45(1), 89–125. <https://doi.org/10.3102/00346543045001089>
- Truong, T. L., Le, H. L., & Le-Dang, T. P. (2020). Sentiment Analysis Implementing BERT-based Pre-trained Language Model for Vietnamese. *Proceedings - 2020 7th NAFOSTED Conference on Information and Computer Science, NICS 2020*. <https://doi.org/10.1109/NICS51282.2020.9335912>
- Vandamme, J. -P., Meskens, N., & Superby, J. -F. (2007). Predicting Academic Performance by Data Mining Methods. *Education Economics*, 15(4), 405–419. <https://doi.org/10.1080/09645290701409939>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 2017-December*.
- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. *EMNLP 2018 - 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Proceedings of the 1st Workshop*. <https://doi.org/10.18653/v1/w18-5446>

- Wang, Y., Wang, C., Li, R., & Lin, H. (2022). On the Use of BERT for Automated Essay Scoring: Joint Learning of Multi-Scale Essay Representation. *NAACL 2022 - 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*. <https://doi.org/10.18653/v1/2022.naacl-main.249>
- Weibull, W. (1951). A Statistical Distribution Function of Wide Applicability. *Journal of Applied Mechanics*, 18(3). <https://doi.org/10.1115/1.4010337>
- Whitehill, J., Mohan, K., Seaton, D., Rosen, Y., & Tingley, D. (2017). MOOC dropout prediction: How to measure accuracy? *L@S 2017 - Proceedings of the 4th (2017) ACM Conference on Learning at Scale*, 161–164. <https://doi.org/10.1145/3051457.3053974>
- Wolff, A., Zdrahal, Z., Herrmannova, D., & Knoth, P. (2014). Predicting student performance from combined data sources. *Studies in Computational Intelligence*, 524, 175–202. https://doi.org/10.1007/978-3-319-02738-8_7
- Xavier, M., & Meneses, J. (2020). A Literature Review on the Definitions of Dropout in Online Higher Education. *EDEN Conference Proceedings*, 0(1), 73–80. <https://doi.org/10.38069/edenconf-2020-ac0004>
- Xavier, M., & Meneses, J. (2022). Persistence and time challenges in an open online university: a case study of the experiences of first-year learners. *International Journal of Educational Technology in Higher Education*, 19(1), 31. <https://doi.org/10.1186/s41239-022-00338-6>
- Xing, W., Chen, X., Stein, J., & Marcinkowski, M. (2016). Temporal predication of dropouts in MOOCs: Reaching the low hanging fruit through stacking generalization. *Computers in Human Behavior*, 58, 119–129. <https://doi.org/10.1016/j.chb.2015.12.007>
- Xu, Z., & Zhu, P. (2023). Using BERT-Based Textual Analysis to Design a Smarter Classroom Mode for Computer Teaching in Higher Education Institutions. *International Journal of Emerging Technologies in Learning (IJET)*, 18(19). <https://doi.org/10.3991/ijet.v18i19.42483>
- Yair, G., Rotem, N., & Shustak, E. (2020). The riddle of the existential dropout: lessons from an institutional study of student attrition. *European Journal of Higher Education*, 10(4), 436–453. <https://doi.org/10.1080/21568235.2020.1718518>
- You, J. W. (2016). Identifying significant indicators using LMS data to predict course achievement in online learning. *Internet and Higher Education*, 29, 23–30. <https://doi.org/10.1016/j.iheduc.2015.11.003>
- Zacharis, N. Z. (2015). A multivariate approach to predicting student outcomes in web-enabled blended learning courses. *Internet and Higher Education*, 27. <https://doi.org/10.1016/j.iheduc.2015.05.002>
- Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education – where are the educators? In *International Journal of Educational Technology in Higher Education* (Vol. 16, Issue 1). <https://doi.org/10.1186/s41239-019-0171-0>



Zhang, H., Cai, J., Xu, J., & Wang, J. (2019). Pretraining-based natural language generation for text summarization. *CoNLL 2019 - 23rd Conference on Computational Natural Language Learning, Proceedings of the Conference*. <https://doi.org/10.18653/v1/k19-1074>

Zhang, X., & Lapata, M. (2017). Sentence simplification with deep reinforcement learning. *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*. <https://doi.org/10.18653/v1/d17-1062>

Zhang, Z., Zhang, Z., Chen, H., & Zhang, Z. (2019). A joint learning framework with BERT for spoken language understanding. *IEEE Access*, 7. <https://doi.org/10.1109/ACCESS.2019.2954766>