





Learning Analytics: Analysis of Methods for Online Assessment

Vito Renò ¹, Ettore Stella ¹, Cosimo Patruno ¹, Alessandro Capurso ², Giovanni Dimauro ²
and Rosalia Maglietta ^{1,*}

¹ National Research Council of Italy, Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing (CNR STIIMA), Via G. Amendola 122 D/O, 70126 Bari, Italy

² Department of Computer Science, University of Bari, Via E. Orabona 4, 70125 Bari, Italy

* Correspondence: rosalia.maglietta@stiima.cnr.it

Abstract: Assessment is a fundamental part of teaching and learning. With the advent of online learning platforms, the concept of assessment has changed. In the classical teaching methodology, the assessment is performed by an assessor, while in an online learning environment, the assessment can also take place automatically. The main purpose of this paper is to carry out a study on Learning Analytics, focusing in particular on the study and development of methodologies useful for the evaluation of learners. The goal of this work is to define an effective learning model that uses Educational Data to predict the outcome of a learning process. Supervised statistical learning techniques were studied and developed for the analysis of the OULAD benchmark dataset. The evaluation of the learning process of learners was performed by making binary predictions about passing or failing a course and using features related to the learner's intermediate performance as well as the interactions with the e-learning platform. The Random Forest classification algorithm and other ensemble strategies were used to perform the task. The performance of the models trained on the OULAD dataset was excellent, showing an accuracy of 95% in predicting the students' learning assessment.



Citation: Renò, V.; Stella, E.; Patruno, C.; Capurso, A.; Dimauro, G.; Maglietta, R. Learning Analytics: Analysis of Methods for Online Assessment. *Appl. Sci.* **2022**, *12*, 9296. <https://doi.org/10.3390/app12189296>

Academic Editor: Manuel Graña

Received: 19 August 2022

Accepted: 13 September 2022

Published: 16 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: learning analytics; online assessment; classification; Moodle

1. Introduction

Assessment is a fundamental part of teaching and learning; it plays a fundamental role in educational policies, especially in attempts to raise educational standards [1]. The term assessment refers to all those actions undertaken by teachers and learners who have as their purpose certain feedback aimed at modifying teaching and learning activities [2]. Evaluation plays an important role in education. In traditional classrooms, it is difficult to know the status of student learning, and teaching evaluations are subject to a certain degree of subjectivity and delay [3].

In the didactic evaluation, it is possible to identify two different typologies: quantitative and qualitative evaluation. The first aims at analytical measurement through the quantification of learning performance. The second is also called formative assessment and aims to improve learning and not to describe it, generating continuous feedback to trigger corrective actions to fill the training gaps. With the advent of e-learning platforms, the concept of formative evaluation has changed, as there is a different context from the classic one in which training usually takes place. In classical evaluation, the evaluation process is reserved for the role of the evaluator, who can be the teacher or a group of teachers of the course, while online, the evaluation is mostly automatic. However, using online e-learning platforms, it is possible to have access to a large amount of data related to the behavior and interactions of teachers and learners during the learning process as well as the final evaluation of a learner. Assessment and evaluation are related, and their relationship could then be exploited, for example, to perform multiple tasks such as predicting the outcome of a learner or studying the variables (features) that led to the outcome prediction.

In fact, Learning Analytics allows students and educators to have access to new tools to improve learning and education processes by allowing them to visualize processes and progress in ways that until now were only accessible to researchers [4]. SoLAR (Society for Learning Analytics Research), the reference research group for Learning Analytics, defines Learning Analytics by emphasizing the central role of the learner as the “measurement, collection, analysis and reporting of data relating to learners and their contexts, in order to understand and optimize learning and the environment in which this occurs” [5]. Learning Analytics uses various techniques tested in similar research areas. There are some classic techniques identified by [6] and some techniques that can be grouped as Social Learning Analytics, according to [7]: statistics, information display, data mining, social network analysis, speech analysis, content analysis, and analysis of inclinations.

In this paper, statistics and data mining have been applied to introduce an innovative model that could be used to automatically assess and evaluate the learning process of a student during online sessions, also using an ensemble modeling strategy. In particular, with reference to the e-learning process, we first defined a certain number of variables of interest (i.e., the features) that were used to build a binary learning classification model. The effective use of these features was demonstrated with a feasibility study of our learning model applied to a publicly available dataset that was used to evaluate the validity of our model. Then, a second contribution was also given as an automated software, in the form of a Moodle plugin, aimed at automatically extract the variables used in the proposed learning model. This way, independently from the specific e-learning course, data in terms of the features needed from our model to predict the outcome of the course can be automatically extracted. The rest of the paper is organized as follows. In Section 2, we report the materials and methods, with a brief focus on the related works, the Moodle platform, and classification models as random forest and ensemble classifiers as well as the introduction of our learning model. In Section 3, we show the experiments and results of this work. In Section 4, we conclude the paper.

2. Materials and Methods

2.1. Related Works

In [8], the author carried out an analysis of student performance using different classifiers. The dataset used is made up of 225 instances made up of ten attributes each from the student college database. Each instance is characterized by intermediate assessments, course attendance, participation in seminars, and a final exam expressed on a scale of three values: Poor, Average, Good; participation in the activities, laboratory experiments, development of the Office Automation project, and participation in the Workshops expressed in two values, Yes and No; the evaluation in the previous semester and the final evaluation expressed with a scale of four values High average, Average, Poor, and Fail. The dataset was analyzed with the help of WEKA software using the following classifiers: Naive Bayes, BayesNet, Iterative Dichotomiser 3 (ID3), C4.5, and Neural Network (NN). WEKA is open-source software developed at the University of Waikato in New Zealand. It contains four applications: explorer, experimental, knowledge flow, and the command line interface (CLI) and contains tools for data preprocessing, classification, regression, and visualization. For the comparison between the various classifiers various measures have been adopted: True Positive Rate (TP Rate), False Positive Rate (FP Rate), and Accuracy. The experimentation showed that the Bayesian Networks performed better than the other classifiers in terms of accuracy obtaining a value equal to 92%.

The authors of [9] analyze the kalboard 360 dataset using WEKA software with the following algorithms: random forest, SVM, Naïve Bayes, Multilayer Perceptron, and DT-J48. The results obtained show that the Multilayer Perceptron achieved the best performance compared to the other classifiers reaching an accuracy value equal to 76.07%.

The authors of [10] carried out a study on student performance prediction by comparing the Random Forest algorithm with the Decision Tree and Naïve Bayes algorithms using

the WEKA open-source software. The comparisons showed that of the three algorithms, the random forest is the one with the highest prediction accuracy at 81%.

The authors of [11] analyzed the results of learners at the end of the quarter in a mixed educational setting. Three different classification algorithms were used for the analysis: logistic regression, decision trees, and random forest. This study applies these three models to make medium-term forecasts and to identify the best one. The objective variables are whether the final exam was passed or not. The analyzed data were collected through an LMS and were weekly assignments, quiz results, and clicks on the contents of the platform. Data from Facebook Graph API were added, which made it possible to collect information regarding the activities of learners on Facebook Live. Comparing the performance of the algorithms previously mentioned, it was found that the Random Forest algorithm was the most performing with 0.81 of F1-Score and 0.83 of Accuracy.

In [12], the authors used time series classification method to predict student's dropout rates using the OULAD dataset. The result of this work is that it is possible to obtain an accuracy value of 84% with only 5% of the data. The highest accuracy reached was 90% with the full dataset. More recently, a review about benchmark studies involving the OULAD dataset was performed in [13], where the authors compare different studies involving learning-based advanced approaches at the state of the art, also based on machine learning and deep learning approaches. Even if these approaches are widespread, the authors of [13] point out current challenges about the level of development of this technology in terms of feature selection, state-of-the-art approaches that are still missing (e.g., encoder–decoder or ensemble modeling), and feature scaling. Finally, in [13], the authors point out that the fact that each study followed different input data preprocessing and that feature scaling approaches led to one of the major problems in the results comparison, namely a varying output performance using the same model.

From the analysis of the literature, it appears clear that machine learning can be useful to perform predictions for learning analytics purposes and is effectively used by re-researchers and stakeholders in the field. However, there is a lack in the definition of a strategy collecting all the variables that could be sampled during e-learning sessions. Solving this issue is an ambitious goal as it should be related to the opportunity of deeply understanding how the learning process is performed, mostly from a student point of view. The learning process is changing and evolving, so the automatic models should be updated as well to better understand how the users interact, both among them and with the e-learning platforms.

2.2. Moodle

Moodle (Modular Object-Oriented Dynamic Learning Environment) is an e-learning platform, i.e., an educational tool, with access and use entirely on the Web, which supports traditional classroom teaching and allows the teacher to publish and make the teaching material of the lessons, to convey communications, to publish information on the course and lessons, and to administer tasks/exercises, tests, and more. Moodle keeps track of all the activities students perform through detailed logs [14]. It records all the single clicks that students make to navigate and has a discrete log viewing system within it. Log files can be filtered by course, participant, day, and activity. The instructor can use these logs to determine who was active in the course, what he did, and when he did it. The logs are not collected as text files but are saved in a relational database. The Moodle database has about 145 correlated tables [15], not all the information present is useful for the elaboration of predictions, classifications, etc. purposes of the latter in a form compatible with the algorithms that will be used.

Moodle provides a set of tools useful for analysis but with limited functionality. These tools are divided into: Reports and Analytics. The activity reports show the number of visits for each course activity, the participation reports show for a particular activity who participated and how many times, and finally, the log reports show for a single user what actions he performed within the LMS. Analytics is a predictive analysis tool, which supports

Machine Learning based on Logistic Regression and Neural Networks algorithms (using TensorFlow) and comes with three models to use: “No teaching”, “Upcoming activities due”, and “Students at risk of dropping out”.

2.3. Classification Methods

2.3.1. Random Forest

The Random Forest (RF) algorithm uses a multiplicity of binary decision trees [16]. Each tree is built using a set of training data, and for each node, a random set of variables considered for the best split is chosen. With reference to Figure 1, a dataset is randomly split and used to build a certain number of decision trees. The final result of the classification is represented by a specific voting step among all the trees that have been built.

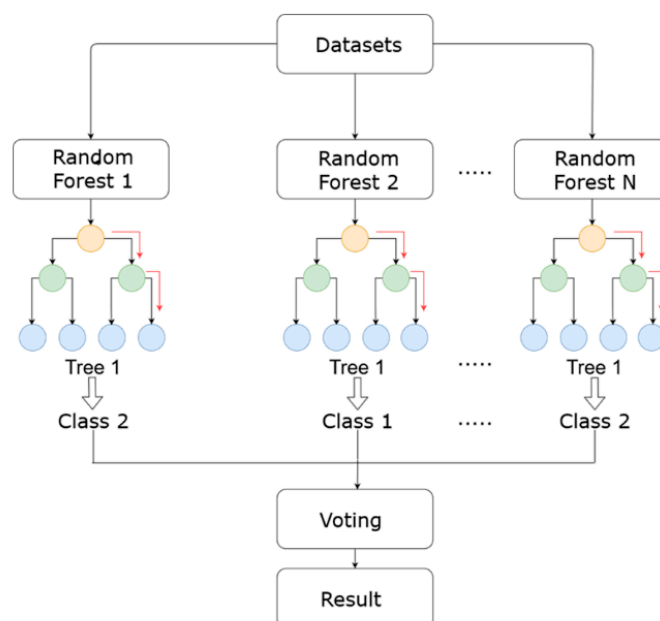


Figure 1. Random forest example scheme.

Random Forest builds a large collection of decorrelated trees and then averages them. Given B as the number of trees to generate, for $b = 1, 2, \dots, B$, the RF algorithms can be briefly described as follows.

- A bootstrap sample Z^* of size n is drawn from the training set
- A random-forest tree T_b is grown from the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size, n_{min} , is reached:
 - Selection of q variables at random forest from the d variables.
 - Choice of the best variable/split-point among q (internal feature selection).
 - Splitting of the node into two daughter nodes.
- Output of the ensemble of trees $\{T_b\}_B$

Given a new point x , let $C_b(x)$ be the class prediction of the b th random forest tree, the prediction of RF on this new sample is given by $y = \text{majority vote } C_b(x)_B$.

Examples of application of Random Forest are given in [17,18].

2.3.2. Ensemble Model

An effective classification strategy consists of using the so-called ensemble classifiers. The main idea behind the ensemble methodology is to aggregate multiple weighted models (among them the before cited Random Forest) to obtain a combined model that outperforms every single model in it. The main theory behind ensemble methods is bias-variance-covariance decomposition. It offers theoretical justification for improved

performance of an ensemble over its constituent base predictors. The key of ensemble methods is diversity, which includes data diversity, parameter diversity, structural diversity, multiobjective optimization, and fuzzy methods [19]. Ensemble classifiers have been successfully used in a wide range of application domains, ranging from the marine biology and environmental protection [20] to the industry setting [21]. In this paper, we have applied this strategy in the learning analytics domain. Specifically, an ensemble classifier is represented by a set of prediction classifiers built using all the predictive variables available in the dataset and comparing the predictive accuracies within the sample of the sets. This optimization method can be used automatically, where a comparison is made between a pool of algorithms chosen by the method based on the type of classification (binary or multiple), but it can also be used by choosing the type of algorithms to be compared (e.g., bag or boost) and by choosing the parameters to be optimized.

3. Experiments and Results

3.1. Dataset

All the experiments performed in this work were conducted on the OULAD benchmark dataset [22]. OULAD is a dataset extracted from the Open University database at Walton Hall, Milton Keynes, UK in 2015. It contains data about courses, students, and their interactions in a Virtual Learning Environment (VLE) in seven selected courses (called modules). The dataset consists of a dump of a relational database, made of data tables connected using unique identifiers, distributed as several comma separated values (.csv) files. A brief overview of the OULAD database schema is reported in Figure 2, where it is immediate to see that three main data categories are reported: the module presentation, in green, devoted to the courses and assessment data storage; the student activities, in purple; and the student demographical information, in yellow.

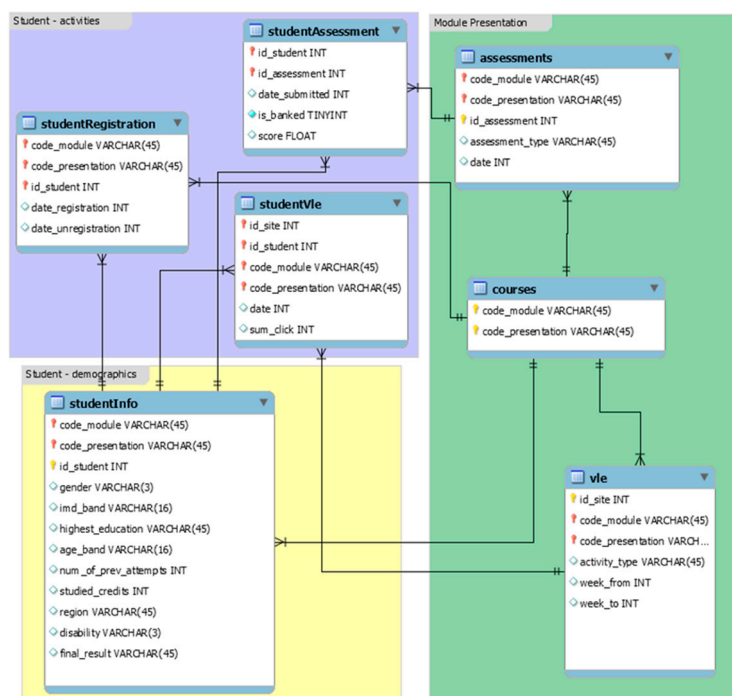


Figure 2. Database schema of OULAD dataset (online available at https://analyse.kmi.open.ac.uk/open_dataset).

The dataset contains data relating to 32,593 students each marked with one of the following labels: “fail”, “withdrawn”, “pass”, and “distinct”. In this study, to perform a binary classification process, the “fail” and “withdrawn” labels were aggregated considering them on the same level, and, therefore, both were labeled as “fail”, same thing

was done with the labels “pass” and “distinct” aggregated under the label “pass”. At the end of the aggregation process, there were 17,208 students labelled as “fail” and 15,385 students labelled as “pass”. From the OULAD data, information about the course, the number of studied credits, the student, the course previous attempts for the student, the assessments (either evaluated by a tutor or a computer), and the examination, as well as the number of interactions with the learning platform are extracted. For each variable, grouped information such as minimum, maximum, sum, and average values over time are reported as well. This way, a total number of 97 features has been automatically grouped and extracted to define the classification problem. A script developed in PHP 7.0 is responsible for extracting such features.

3.2. Evaluation Metrics

The statistical metrics used to evaluate the performance of the classifiers are here reported. In the context of binary classification, it is usual for a prediction of the system to be called “positive” or “negative”. In our case study, the label “positive” refers to a student who has passed the course, while the “negative” label is associated to a failure. Once positive and negative classes have been defined, it is possible to understand if the prediction made by a classifier is correct or not by computing:

- false positive (FP): student labeled as fail and wrongly classified as pass;
- false negative (FN): student labeled as pass and wrongly classified as fail;
- true positive (TP): student labeled and correctly classified as pass;
- true negative (TN): student labeled and correctly classified as fail.

The set of these indicators is typically organized in a matrix called the confusion matrix. Based on these parameters, three main metrics were extracted:

- Accuracy: measures the relationship between correct predictions and the total number of instances evaluated. Let n be the number of instances evaluated, with $n = TP + FP + TN + FN$

$$\text{Accuracy} = \frac{(TP + TN)}{n}$$

- Sensitivity (Rate of true positives): measures the proportion of positively identified positives (i.e., the proportion of those who have some condition (affected) who are correctly identified as affected by the condition).

$$\text{Sensitivity} = \frac{TP}{(TP + FN)}$$

- Specificity (Rate of true negatives): measures the proportion of negatives that are correctly identified.

$$\text{Specificity} = \frac{TN}{(FP + TN)}$$

3.3. Classification with Random Forest

In this experiment, the Random Forest model was used with the aid of the MATLAB method called “TreeBagger”. The model was tested by varying two parameters: number of Cross-Validation and number of trees generated. The type of Cross-Validation applied for this first model was the CV Holdout with a rate of 70–30%, meaning that the 70% of data was used for model training and the 30% was used for the testing phase. To validate the model, the Holdout Cross-Validation technique was applied with a number of iterations equal to 50. The number of trees of RF varied as 1, 5, 10, 100, 1000, and 10,000. Table 1 shows the experimental results obtained.

Table 1. Accuracy, Sensitivity, and Specificity of Random Forest classifiers using the OULAD dataset. Number of trees represents the number of trees generated by Random Forest. The metrics are averaged with respect to the number of cross-validation (50).

Number of Trees	Metrics		
	Accuracy (%)	Sensitivity (%)	Specificity (%)
1	90.23	90.03	90.43
5	92.11	93.04	91.19
10	92.90	94.24	91.55
50	93.45	95.28	91.63
100	93.89	95.95	91.82
1000	94.14	95.28	91.63
10,000	94.32	96.79	91.85

The Random Forest model that obtained the best results is the one with the highest number of trees, namely 10,000, which scored 94.32% for the Accuracy metric, 96.79% for the Sensitivity, and 91.85% for the Specificity, respectively. However, it is worth noting that in any case the metrics values are above 90% and that, with a sensibly lower number of trees, 1000 instead of 10,000, results are almost comparable. This suggests that 1000 trees can be considered as a good parameter to tune up the algorithm. Finally, as to the Specificity, the score is almost 91% from 5 trees to the highest number.

3.4. Classification with Other Ensemble Model

A second experiment has been performed to evaluate the classification with other ensemble models that can be performed in MATLAB using the method called “fitcensable”, which generates an optimized classification model as specified by the user. More specifically, when it is set in automatic mode, the method applies a 5-fold CV and identifies the algorithm with the best performance while the values of the internal variables are updated. Moreover, “fitcensemble” allows the user to select the type of algorithms to use and to optimize the number of learning cycles, that refers to the number of weak learners trained (i.e., the number of trees used), and the maximum number of splits.

First, an ensemble classifier was trained using the bag ensemble aggregation method. This means that the algorithm uses bagging with random predictor selections at each split (random forest). Similar to the previous experiment, the cross-validation strategy was the holdout with 70%–30% ratio between training and test set, repeated 50 times. Table 2 shows the obtained results:

Table 2. Accuracy, Sensitivity, and Specificity of the bagging ensemble strategy.

Metrics	Values (%)
Accuracy	95.23
Sensitivity	97.98
Specificity	92.76

A first comparison with the previous experiment shows that the values of Accuracy, Sensitivity, and Specificity in this case are greater than the previous ones, showing a difference of 0.91% in accuracy, 1.19% in sensitivity, and 0.89% in specificity. A final remark should be given about the optimal parameters that were computed by the algorithm. In more details, the ensemble strategy is built by evaluating for a certain number of times a classification algorithm updating its parameters, and every time the results are better with respect to the previous runs, the best classifier is updated accordingly.

Finally, another ensemble was trained by setting the method to be a boosting algorithm with decision tree learners. The training and cross-validation strategy was the same as the previous experiments. Table 3 recaps the results obtained:

Table 3. Accuracy, Sensitivity, and Specificity of the boosting ensemble strategy.

Metrics	Values (%)
Accuracy	95.18
Sensitivity	97.40
Specificity	93.19

The results obtained in this last experiment are similar to those obtained with the previous one with slightly lower values in terms of accuracy and sensitivity and a slightly higher specificity value. Moreover, in this experiment, we notice a greater balance between the sensitivity and specificity metrics.

3.5. Learning Features

The learning features proposed in this work are based on the assumption that a certain number of variables can be used and combined by machine learning techniques with the aim of understanding the learning process. In general, our underlying hypothesis is that the classifier should take the users' interactions with the e-learning instruments over time into account instead of simply considering a quiz result in terms of a score achieved. A similar approach has been defined in [23], where the authors suggest consideration of the whole academic achievements of a learner over time to improve the prediction results for their learning model. For this reason, variables such as the number and type of documents present in a course (e.g., slides, images, interactive material, etc.) as well as the number and type of interactions of the user with the platform are extracted. Even if the preliminary results have been presented with reference to the OULAD dataset, to ensure an effective and automatic information retrieval using custom e-learning courses based on the Moodle platform, a custom software plugin has been developed in PHP 7.0 for this purpose. This software plugin could be extremely useful to reduce bias that could be experienced in case of manual data extraction from custom databases. Moreover, the goal was to create a plugin useful for extracting the data necessary for the analysis from any type of Moodle course. With reference to the feature extraction performed for OULAD dataset, reported in Section 3.1, the following macrofeatures have been identified to be extracted from Moodle:

- TotDocuments: Total number of documents present in the course.
- TotResouces: Total number of resources present in the course.
- Userid: User id.
- NLogin: Number of logins made by the user.
- SessionTime: The length of stay on the platform is indicated for each login.
- NSyncInteraction: Number of synchronous interactions made by the user.
- SyncInteractionTime: Duration of synchronous user interactions.
- NAsyncInteraction: Number of asynchronous interactions made by the user.
- NCompleted Documents: Number of documents completed by the user.
- NCompleted Resources: Number of resources completed by the user.
- QuizId: ID of the quiz carried out by the user.
- Attempts: Attempts made at the same quiz by the user.
- Grades: Grades obtained for each attempt.
- DurationTestPassed: Duration of the test inherent in the successful attempt.
- NAccess: Number of test accesses for each attempt.

It is worth noting that for these macrofeatures, aggregate functions such as minimum, maximum, sum, and average values over time are extracted to be comparable to the features described in Section 3.1. The plugin generates a file for each course on the platform in the

“admin/cli” folder of Moodle. The generated files can be directly used as the input for subsequent classification algorithms. The extraction process is scheduled over time, as it relies on Moodle’s Cron service (more specifically it runs every week).

4. Conclusions

In this paper, a preliminary study about the evaluation of the learning quality (as perceived by the learner) is provided. The models that we developed are able to predict a binary variable—namely a pass/fail evaluation—by combining multiple features that are related to the e-learning platform experience of the learner, so at this stage, the perception refers to whether a student passes the course or not. The work has been focused to propose a learning analytics model and identify a set of parameters for performing automatic learners’ assessment. We focused on supervised learning techniques developing ensemble classifiers and testing them on a benchmark dataset, namely OULAD, using specific learning features that have been grouped and extracted from the dataset. Our learning model takes different variables into account, ranging from the learner’s score to his/her interactions with the platform resources as well as with other learners, showing its effectiveness in the prediction of the final assessment of the learner. The results shown in this paper suggest the applicability of the approach on a larger scale. To this end, we also developed a Moodle plugin that is able to periodically extract the input features of our model from the e-learning platform in an automatic way for training and evaluation purposes. Future directions of this research will be devoted to larger experimentations directly on Moodle platforms with a high number of active learners aimed at evaluating their performance over time and trying to give useful feedback about the quality of the learning process itself. In addition, having access to structured data for each student and for each subject will also enable the exploitation of individual analytics (or cluster of analytics) for multiple purposes such as better understanding whether a student passes or fails/withdraws or evaluative ones to influence how a curriculum is designed or delivered. As soon as a greater number of data will be available, other methodologies could also be developed and tested to integrate and improve this approach (e.g., trying to distinguish and identify the factors that lead to a failure separately from the ones that lead to withdrawal), such as deep learning techniques.

Author Contributions: Conceptualization, V.R. and R.M.; methodology, V.R. and R.M.; software, A.C.; validation, G.D. and C.P.; writing—original draft preparation, V.R., A.C. and R.M.; writing—review and editing, all authors; supervision, E.S.; funding acquisition, E.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work is within the project STELLE—Satellite Technology to Enabling new Learning and Lessons Environment (<https://business.esa.int/projects/stelle>).

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank Protom Group S.p.A. for their support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pachler, N.; Mellar, H.; Daly, C.; Mor, Y.; Wiliam, D.; Laurillard, D. *Scoping a Vision for Formative E-Assessment: A Project Report for JISC*; Joint Information Systems Committee (JISC), Institute of Education: London, UK, 2009; Volume 40, pp. 1–128. Available online: <http://www.jisc.ac.uk/media/documents/projects/scopingfinalreport.pdf> (accessed on 18 August 2022).
2. Black, P.; Wiliam, D. Inside the Black Box: Raising Standards through Classroom Assessment. *Phi Delta Kappan* **2010**, *92*, 81–90. [CrossRef]
3. Dimauro, G.; Scalera, M. The Educational Cloud, Problems and Perspectives. In Proceedings of the 20th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI 2016), Orlando, FL, USA, 5–8 July 2016; Volume 14, pp. 34–40.
4. Simon, K.N.I.G.H.T.; Shum, S.B. Theory of Learning Analytics. Society for Learning Analytics Research (SoLAR). 2017. Available online: <https://www.solaresearch.org/publications/hla-17/hla17-chapter1/> (accessed on 18 August 2022).

5. Siemens. Open Learning Analytics: An Integrated & Modularized Platform. Society for Learning Analytics Research (SoLAR). 2011. Available online: <https://www.solaresearch.org/core/open-learning-analytics-an-integrated-modularized-platform/> (accessed on 18 August 2022).
6. Chatti, M.A.; Dyckhoff, A.L.; Schroeder, U.; Thüs, H. A reference model for learning analytics. *Int. J. Technol. Enhanc. Learn.* **2012**, *4*, 318–331. [[CrossRef](#)]
7. Ferguson, R.; Buckingham Shum, S. *Towards a Social Learning Space for Open Educational Resources*; IGI Global: Hershey, PA, USA, 2012; pp. 309–327. Available online: <http://www.igi-global.com/book/collaborative-learning-open-educational-resources/59714> (accessed on 18 August 2022).
8. Almarabeh, H. Analysis of Students' Performance by Using Different Data Mining Classifiers. *Int. J. Mod. Educ. Comput. Sci.* **2017**, *9*, 9–15. [[CrossRef](#)]
9. Jalota, C.; Agrawal, R. Analysis of Educational Data Mining using Classification. In Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 14–16 February 2019; pp. 243–247. [[CrossRef](#)]
10. Ajay, P.; Pranati, M.; Ajay, M.; Reena, P.; BalaKrishna, T. Prediction of student performance using random forest classification technique. *Int. Res. J. Eng. Technol.* **2020**, *7*, 4.
11. Hung, H.-C.; Liu, I.-F.; Liang, C.-T.; Su, Y.-S. Applying Educational Data Mining to Explore Students' Learning Patterns in the Flipped Learning Approach for Coding Education. *Symmetry* **2020**, *12*, 213. [[CrossRef](#)]
12. Haiyang, L.; Wang, Z.; Benachour, P.; Tubman, P. A Time Series Classification Method for Behaviour-Based Dropout Prediction. In Proceedings of the 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT), Mumbai, India, 9–13 July 2018; pp. 191–195. [[CrossRef](#)]
13. Alhakbani, H.A.; Alnassar, F.M. Open Learning Analytics: A Systematic Review of Benchmark Studies Using Open University Learning Analytics Dataset (OULAD). In Proceedings of the 2022 7th International Conference on Machine Learning Technologies (ICMLT), New York, NY, USA, 11 March 2022; pp. 81–86.
14. Rice, W. *Moodle E-Learning Course Development: A Complete Guide to Create and Develop Engaging E-Learning Courses with Moodle*, 3rd ed.; Packt Publishing: Birmingham, UK, 2015.
15. Romero, C.; Ventura, S.; García, E. Data mining in course management systems: Moodle case study and tutorial. *Comput. Educ.* **2008**, *51*, 368–384. [[CrossRef](#)]
16. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
17. Maglietta, R.; Amoroso, N.; Bruno, S.; Chincarini, A.; Frisoni, G.; Inglese, P.; Tangaro, S.; Tateo, A.; Bellotti, R. Random Forest Classification for Hippocampal Segmentation in 3D MR Images. In Proceedings of the 2013 12th International Conference on Machine Learning and Applications, Miami, FL, USA, 4–7 December 2013; Volume 1, pp. 264–267. [[CrossRef](#)]
18. Inglese, P.; Amoroso, N.; Boccardi, M.; Bocchetta, M.; Bruno, S.; Chincarini, A.; Errico, R.; Frisoni, G.B.; Maglietta, R.; Redolfi, A.; et al. Multiple RF classifier for the hippocampus segmentation: Method and validation on EADC-ADNI Harmonized Hippocampal Protocol. *Phys. Med.* **2015**, *31*, 1085–1091. [[CrossRef](#)] [[PubMed](#)]
19. Ren, Y.; Zhang, L.; Suganthan, P.N. Ensemble Classification and Regression-Recent Developments, Applications and Future Directions. *IEEE Comput. Intell. Mag.* **2016**, *11*, 41–53. [[CrossRef](#)]
20. Chang, Y.-S.; Abimannan, S.; Chiao, H.-T.; Lin, C.-Y.; Huang, Y.-P. An ensemble learning based hybrid model and framework for air pollution forecasting. *Environ. Sci. Pollut. Res.* **2020**, *27*, 38155–38168. [[CrossRef](#)] [[PubMed](#)]
21. Li, S.; Feng, L.; Ge, Y.; Zhu, L.; Zhao, L. An Ensemble Learning Method for Robot Electronic Nose with Active Perception. *Sensors* **2021**, *21*, 3941. [[CrossRef](#)] [[PubMed](#)]
22. Kuzilek, J.; Hlosta, M.; Zdrahal, Z. Open University Learning Analytics dataset. *Sci. Data* **2017**, *4*, 170171. [[CrossRef](#)] [[PubMed](#)]
23. Ahmed, A.B.E.D.; Elaraby, I.S. Data Mining: A prediction for Student's Performance Using Classification Method. *World J. Comput. Appl. Technol.* **2014**, *2*, 43–47. [[CrossRef](#)]