applied sciences

MDPI

Article

Assignments as Influential Factor to Improve the Prediction of Student Performance in Online Courses

Aurora Esteban, Cristóbal Romero and Amelia Zafra *

Department of Computer Science and Numerical Analysis, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Cordoba, 14071 Cordoba, Spain; aestebant@uco.es (A.E.); cromero@uco.es (C.R.)

* Correspondence: azafra@uco.es

Abstract: Studies on the prediction of student success in distance learning have explored mainly demographics factors and student interactions with the virtual learning environments. However, it is remarkable that a very limited number of studies use information about the assignments submitted by students as influential factor to predict their academic achievement. This paper aims to explore the real importance of assignment information for solving students' performance prediction in distance learning and evaluate the beneficial effect of including this information. We investigate and compare this factor and its potential from two information representation approaches: the traditional representation based on single instances and a more flexible representation based on Multiple Instance Learning (MIL), focus on handle weakly labeled data. A comparative study is carried out using the Open University Learning Analytics dataset, one of the most important public datasets in education provided by one of the greatest online universities of United Kingdom. The study includes a wide set of different types of machine learning algorithms addressed from the two data representation commented, showing that algorithms using only information about assignments with a representation based on MIL can outperform more than 20% the accuracy with respect to a representation based on single instance learning. Thus, it is concluded that applying an appropriate representation that eliminates the sparseness of data allows to show the relevance of a factor, such as the assignments submitted, not widely used to date to predict students' academic performance. Moreover, a comparison with previous works on the same dataset and problem shows that predictive models based on MIL using only assignments information obtain competitive results compared to previous studies that include other factors to predict students performance.

Keywords: Multiple Instance Learning; educational data mining; OULAD; virtual learning system; predicting performance

Citation: Esteban, A.; Romero, C.; Zafra, A. Assignments as Influential Factor to Improve the Prediction of Student Performance in Online Course. *Appl. Sci.* **2021**, *11*, 10145. https://doi.org/10.3390/app112110145

Academic Editor: Lidia Jackowska-Strumillo

Received: 25 July 2021
Accepted: 23 October 2021
Published: 29 October 2021

1. Introduction

The popularization of Internet access and the advances in the exploration of digital resources have led to a growing interest in distance education. This education has as main advantages the accessibility (students can follow a course from anywhere in the world) and the flexibility (students can fit their learning around their daily routine) [1]. The current distance studies could not be understood without a digital platform that provides fundamental features like the publication of the contents of the course, a channel to maintain professor-students communication or the tools to keep a control of the student evolution. These systems, called Virtual Learning Environments (VLEs), include course content delivery instruments, quiz modules and assignment submission components, among other functionalities [2]. In addition, VLEs are very useful to control the student involvement in the course, since all his/her activity is recorded in log files that can be analyzed [3].

Even though the history of distance courses is too recent, they have experimented a high expansion, with Massive Open Online Course (MOOCs) as the most popular

example. This new format is opening a widespread investigation, due to its differences with respect to traditional face-to-face higher education. Among these features, an open environment regardless of the location of students stands out, as well as a higher number of enrollments [4]. This also implies an important growth in the number of dropouts and academic failures [5]. In this context, the prediction of student success according to their work collected by VLE's system has become an essential task to be able to discover the main features that describe to the students that pass satisfactorily a course. Thus, some works analyze the impact of the chosen learning platform [6,7], others study the effectiveness of the student-instructor interaction in this engagement [8], while other works have focused on the dynamic adaptation of the e-learning system to the current level of knowledge of each student, based on the interaction with the exercises of the course [9] or prior knowledge and other social factors of students [10].

In this context, this work explores a little used factor to predict student success in distance learning analyzing how this information should be treated to extract all its potential. Specifically, the factor proposed is assignment information, understood as the tasks submitted by the students throughout the online course. The use of information about assignments is not extended as essential factor to determine the students' performance [1,4,11]. Preliminary, it could be estimated that delivered assignments may help predict the student performance more effectively than the number of accesses or clicks on the course resources, that it is the most used factor. Probably, the lower use of assignments information may be due to a combination of the little number of users who complete assignments, in relation to the total enrollments, and the substantial variation across courses in the assignments scheduled. In this context, our work proposes to use assignments information from a flexible data representation perspective based on Multiple Instance Learning (MIL). This learning framework introduced by Dietterich et al. [12] is considered an extension of supervised traditional learning focused on weakly labeled data. MIL could make a better use of the information provided by submitted assignments to predict the impact of students' achievement.

In the field of educational data mining there are not many public datasets due to the sensitivity of the data being worked with. This makes it difficult to compare different proposals since the data used by each study are usually so different to be compared. In this context, it is of great relevance the Open University Learning Analytics Dataset (OULAD) [13] availability, one of the few existing public datasets in the field. OULAD collects a large number of student data from an important distance university during two academic years, including demographic data, student interactions with the VLE and assignments submitted. Many works have used it to predict students' performance but mainly centered in clicks activity, as it is discussed in related work section. This work uses OULAD to obtain assignments information, adapt the representation to the MIL paradigm and store it in ARFF format to work with a popular framework for data mining called Weka. Thus, following the open access philosophy, these files have been made public online for the scientific community that wants to continue the line of this study.

Summarizing, this work carries out an exhaustive study to determine the relevance that information about assignments has to predict the student's performance. Specifically, this work addresses the following research questions:

1. How should the information about assignments be represented? Previous works in distance learning use a classical representation based on single instances. However, each course has different type and number of assignments, and these are submitted by few students, which suggests a high sparsity in the data. Representation should be adapted to this environment so that machine learning algorithms can perform well. We propose to use an optimized representation based on MIL able to adapt to the specific information available for each student.
2. Are machine learning algorithms affected by the way that assignments are represented? It is analyzed a wide set of machine learning algorithms using two different representations of assignments information: representation based on single instances

(it is used in previous studies) and based on MIL (the representation that it has been proposed in the previous step). A significant performance difference between the same algorithms using both representations shows the relevance of an appropriate representation so that assignments can be considered a very influential factor for predicting students' performance.

3. Is information about assignments a relevant feature to predict the student performance? The accuracy in predicting student performance using MIL is compared with previous studies that use different factors such as demographic features and interactions on VLEs to address the same problem. Algorithms using only information about submitted assignments reach competitive results achieving better accuracy in relation to the previous works that predict academic performance using other factors provided in the same dataset. This justifies the relevance of assignments to predict students' performance, if it is represented appropriately.

This paper is organized in following sections. Section 2 presents a briefly introduction to MIL and other concepts of background for this study. In Section 3, a review of related work for solving students' success prediction tasks in distance education is presented. Moreover, this section briefly introduces MIL and its application to the educational environment. Section 4 addresses an in-depth analysis of problem representation and the available information. In Section 5, it is presented the experimentation carried out and the obtained results. Finally, Section 6 draws some conclusions and proposes some ideas for future work.

2. Background

This section presents a basic background of concepts to understand the rest of the work. On the one hand, a brief introduction to MIL is carried out. On the other hand, it is presented the description of the algorithms that will be used in the comparative study from the traditional and MIL perspectives.

2.1. Multiple Instance Learning

Multiple Instance Learning (MIL) was introduced by Dietterich et al. [12] to represent complicated objects [14]. Its inherent capacity to represent ambiguous information allows an efficient representation of different types of objects, such as alternative representations or different views of the same object [15], compound objects formed by several parts [16] or evolving objects composed of samples taken at different time intervals [17].

The main characteristic of MIL is its input space representation: patterns are represented as bags which can contain a variable number of instances. In a supervised learning environment based on multi-instance, each bag or pattern has a label, however there is no information about the instance label. Thus, the hypothesis that relates each instance with each bag depends on the type of representation used. One of the most used is known as standard MI assumption, defined by Dietterich et al. [12]. This hypothesis determines that a bag represents a specific concept whether at least one of its instances represents the desired concept to learn, and the bag does not represent the concept whether none of its instances represent it. However, with the application of MIL to more domains, different assumptions have been proposed [18]. Formally, in a traditional machine learning setting, an object M can be represented by a feature vector $V(M)$ associated with a label $f(M)$, $(V(M), f(M))$. However, in multiple instance learning setting, each object M may have a variable number n of instances m_1, m_2, ..., m_n, and each instance has an associated features vector $V(m_i)$, thus the complete training object M is represented as $(\{V(m_1), V(m_2), \ldots, V(m_n)\})$ associated with a label $f(M)$, $(\{V(m_1), V(m_2), \ldots, V(m_n)\}, f(M))$.

2.2. Supervised Data Mining Techniques for Predicting Students' Performance

Predicting students' performance has been addressed from a wide range of popular methods within the field of supervised data mining [2]. There is a special attention to those models that are explainable, since they allow to identify the most determining factors in

the result, i.e., student demographic information, VLE activity, etc. Thus, most popular methods for predicting students' performance are those based on decision trees [11]. These methods offer one of the most intuitive solutions: nodes in the decision tree involve specific predictive factors, and leaf nodes give a classification that applies to all students that reach that leaf. With similar points, rule-based methods offer a solution composed by an antecedent that presents several logical expressions and a consequent that gives the outcome for students covered by the rule. Bayesian methods and logistic regression are among most popular methods too, since they offer predictions based on likelihood of classes where it is possible to determinate the influence of each factor in the result. Support vector machines (SVM) are relatively popular as well, with an approach based on finding the maximum-margin hyperplane in the factor space that gives the separation between types of students. Artificial Neural Networks (ANN) are non-linear models composed of units organized in layers that transmit and transform an input, i.e., the student information, to the end of the network to provide a prediction. These models are less popular [11] because of their lack of explainability, although, on the other side, they tend to be more accurate in their predictions.

3. Related Work

Although VLEs have been used in traditional education for several years, their application to distance education has important particularities. Thus, distance education usually has higher number of enrolled students, more diverse demographic characteristics and, in general, a lower motivation level. These characteristics cause more academic failures and higher dropout rates. In this context, the task of predicting student success in distance education is particularly challenging [3,4].

This section presents a review of previous works and more specifically works that use OULAD. As it has been commented in introduction section, this dataset has had a notable relevance in EDM. In addition, it is addressed a review of the application of MIL framework in education.

3.1. Predicting Student Success in Distance Higher Education

Predicting student' performance in higher education is a problem that has attracted great attention [1,11,19]. Due to the rise of the VLE, online activities and the increase in log data generated by these environments that can be processed with machine learning techniques in order to detect at-risk students, measure the effectiveness of the e-learning system or give an idea of the success of the academic institution. In this context, several student background factors, previous academic record, or activity during the course can be selected to measure his/her engagement, and therefore, the chances of success in the course. According to [19], the most influential factors are the prior academic achievement (44%) and the demographic information (25%). In [11], these factors are also the most common, but they also include e-learning activities (25%) in the top-3 ranking. The e-learning category includes different statistics like number of logins, assignments or quizzes done. However, the number of clicks on the course resources is the most used factor by far in this category.

Recent proposals for predicting students' performance in distance higher education include several works such as [20]. This work combines demographic, assignments and clicks information with information about interaction with video of the recorded classes. The case of [21] also explores three-based methods combining assignments submission and clicks information. In [22], clicks information is explored, but from a frequency perspective rather than number of clicks during the course. In [23], it is explored a novel proposal based on graphical visualization of the logit leaf model that combines demographic, number of clicks and submitted assignments. The case of weakly labeled data in the student record is also addressed from an active learning perspective [24], from a semi-supervised learning approach [25] and from number of clicks. However, all these proposals work with sensible data that can not been published, and each one considers distinct demographic or assignments attributes, so it is difficult to compare proposals and results.

This study uses for evaluating results the OULA dataset or OULAD [13]. It is one of the few existing open datasets about learning analytic and educational data mining. Moreover, it is collected from a real case of study, specifically at the Open University (https://www.open.ac.uk/, accessed on 23 October 2021), the largest institution of distance education in United Kingdown and one of the most important worldwide, with around 170,000 students per year and a wide range of degrees, as well as free courses under its platform Open Learn (https://www.open.edu/openlearn/, accessed on 23 October 2021). The dataset contains information of 32,593 students and 7 courses in their different semesters. It is focused on students, aggregating their demographic data, information about their course enrollments, number of daily accesses done to course resources (clickstream), and records about assignments submitted to the VLE (referred as their assessments) during a course. Due to the characteristics of this dataset and the large amount of information, some literature works have referenced it as MOOC data [26–29]. Specific dataset details are addressed in-depth in Section 4.

An open dataset with these characteristics implies the possibility of having a common framework where different authors can compare their studies with previous works. In this context, although it is a recent dataset (published in 2017), it has reached a high relevance in the field, counting with more than 20 works to date that use it to study the problem of predicting the academic performance. Table 1 summarizes the main characteristics of these previous proposals, taking into account the purpose of the study, the criteria used and the algorithm proposed (or the main one among proposals).

Considering the factors used to carry out the prediction tasks in OULAD, it can be observed slight differences in the most used factors with respect to the previous general work. Thus, a 39% of studies use the number of accesses to resources (clickstreams) [26,29–35], while a 25% of studies combine this information with demographic data from the students [27,28,32,36–39]. Focusing solely on assignment information, only one study [40] uses exclusively this factor. Concretely, it considers assignments as an important factor to predict the student's performance. However, this study has important limitations, like the fact of analyzing only two courses of a total of seven available. The other studies that use assignment information, a 30% of works, use this factor together with the rest of sources sources [3,39,41–43]. Thus, the real relevance of this factor in the final prediction cannot be analyzed.

Regarding the purpose of the different works, under the main task of predicting student performance, it can be found that the majority of studies pretend to predict whether the student will pass or fail a course [3,27,30–32,37–45]. Other approaches focus on the dropout rate [26,29,32,33], while others follow an early prediction study [33,35,36,46]. It is also notable that most studies distinguish among courses or even presentations to make these predictions.

With regard to the algorithms used, it can be appreciated that different supervised learning techniques have been used, mainly trees-based methods [28,30,31,33,36,40,44,46], although Bayesian methods have also been applied [38], as well as support vector machines [27], generative methods based on sequences [26,32] and neural networks or deep learning [3,34,35,39,41–43,45].

This work explores how only information on assignments improves the prediction of academic achievement. The purpose is to show that less data can be used more efficiently to obtain competitive results. For this aim, a study is carried out including all OULAD courses, as well as all specific characteristics of assignments in each course with different representations. The study is conducted over a set of different machine learning algorithms belonging to different paradigms in order to provide a comparison as representative as possible.

Table 1. Comparison between proposals that use OULAD.

Work	Algorithm	Criteria	Prediction
[28]	Decision Tree	Demographic data. Number of clicks per day. Assignments data.	Final outcome. All courses together.
[40]	Decision Tree	Assignments data.	Final outcome: Fail vs. all. In courses CCC and FFF.
[44]	Decision Tree	Demographic data.	Final outcome, excluding Withdraw. In course AAA, per presentation.
[30,31]	J48	Number of clicks per resource.	Final outcome binarized in Pass+Distinction/Fail+Withdraw. All courses separately.
[33]	J48	Number of clicks per resource.	Engagement to the course: a combination of the first assignment score, course final result and total number of clicks.
[47]	Random Forest	Demographic data. Number of clicks per day. Assignments score.	Final outcome binarized in Pass+Distinction/Fail+Withdraw. All courses together. At different percentages of course length
[36,46]	XGBoost	Demographic data. Statistics over of clicks until the first assignment of the course.	Deadline compliance. In courses BBB, DDD, EEE, FFF, only last presentation
[38]	Naive Bayes	Demographic data. Total number of clicks, only in web page resource.	Final outcome, only Pass or Fail. All courses together.
[27]	Support Vector Machine	Demographic data. Number of clicks per day.	Final outcome binarized in Pass+Distinction/Fail+Withdraw. All courses together.
[29]	Gaussian Mixture Model	Number of clicks and number of sessions per resource and time-interval.	Final outcome: Withdraw vs. all. Course BBB. At different intervals of the course.
[37]	Dynamic Incremental Semi-Supervised Fuzzy C-Means	Demographic data. Number of clicks per resource. Assignments average score and number of submissions.	Final outcome binarized in Pass+Distinction/Fail+Withdraw. Course DDD.
[26]	Time Series Forest	Number of clicks per resource and day, only in 3 resources.	Final outcome: Withdraw vs. all. All courses and presentations separately.
[32]	Markov Chains	Number of clicks per week in planned and non-planned activities.	Final outcome: Withdraw vs. all. Course FFF, one presentation.
[3]	Artificial Neural Network (ANN)	Demographic data. Number of clicks per assignment. Assignment score.	Regression of final score. Course DDD, by presentations.
[41]	Deep Artificial Neural Network	Demographic data. Number of clicks. Assignments data.	Final outcome: Fail vs. all. In all courses. At different quarties of course.
[42]	Joint Neural Network Model	Demographic data. Number of clicks per resource and day.	Final outcome, only Pass or Fail. Courses BBB, CCC, FFF, one presentation.
[39]	Recurrent Neural Network	Demographic data. Number of clicks per week and resource. Assignment data.	Final outcome binarized in Pass+Distinction/Fail+Withdraw. All courses together. At different weeks.
[43]	Convolutional and recurrent deep model	Demographic data. Number of clicks. Assignment score.	Final score discretized in six ranges. Course AAA.
[45]	Up-sampling based on Adversarial Network + ANN	Number of clicks per resource and course quartiles.	Final outcome, only Pass or Fail. All courses together.
[34]	LSTM	Number of clicks per week of 25 first weeks.	Final outcome: Withdraw vs. all.
[35]	LSTM	Number of clicks per week.	Final outcome, only Pass or Fail. All courses together.

3.2. MIL in Educational Data Mining

MIL has been used in a wide range of application domains, including classification, regression, ranking and clustering tasks [14]. This framework has experienced a growing

interest to represent problems because of its characteristics in data representation. MIL can naturally adapt to complex problems and it allows to work with weakly labeled data. Prediction of the student's performance from VLE logs has also been addressed with this learning approach [16]. This previous research is set in a different context of traditional e-learning courses. Thus, it is taught in combination with face-to-face classes and it uses different factors in the study. However, it can be considered as an example of MIL efficiency to represent educational data mining problems. From another perspective in [48], it is shown a tool to discover relevant e-activities for learners using MIL.

4. Materials and Methods

In this section, information on assignments in Open University Learning Analytics Dataset (OULAD) [13] is analyzed. It is an anonymized, public and open dataset supported by Open University of United Kingdom. It maintains information about courses, students and their interactions with VLE.

In this section, the original structure of OULAD is analyzed first; secondly, the problem of predicting student's performance from the activity associated to his/her submitted assignments is discussed and, finally, it is addressed the representation based on MIL and the main differences with respect to traditional representation.

4.1. Information Analysis of OULAD

The original source of OULAD has been published by Kuzilek et al. at (https://analyse.kmi.open.ac.uk/open_dataset, accessed on 23 October 2021). It contains 7 distance learning courses (called *modules*), all of them taught at the Open University in several semesters during the years 2013 and 2014 (called *presentations*). The courses consider different domains and difficulties. Thus, courses AAA, BBB and GGG belong to Social Sciences domain and courses CCC, DDD, EEE and FFF to Science, Technology, Engineering and Mathematics (STEM). Concerning to difficulty levels, AAA is a level-3 course, GGG is a preparatory course, and the rest are level-1 courses [36]. Each course has several resources on the VLE used to present the contents of the course, one or more assignments that mark the milestones of the course and a final exam. In total, it contains records of 32,593 students. There is demographic information, such as their gender, region or age band. There is information related to their enrollment in the courses, such us number of previous attempts or the final mark obtained in the course. Also, there is information related to their activity during the courses. This information includes interactions with the resources in the VLE, number of clicks, and the submitted assignments during the course.

An overview of the course structure can be seen in Figure 1. Students can register in several courses during a semester. Moreover, courses are repeated in different years (they have different editions). The content of a course is usually available in VLE a couple of weeks before the official course start. The course assignments are defined as their *assessments* whose purpose is to have a control of the student's evolution. During the presentation of the course, students' knowledge is evaluated by means of assignments which define milestones. Two types of assignments are considered: Tutor Marked Assessment (TMA) and Computer Marked Assessment (CMA). If the student decides to submit an assignment, the VLE collects information about the date of submission and the obtained mark. By contrast, if a student doesn't submit the assignment, no record is stored. At the end of a presentation, a student enrolled in a certain course takes a final exam and achieves a final mark. This mark can take 3 different values: *pass*, *distinction* or *fail*. Additionally, if the student doesn't carry out this exam, it will be considered that he/she doesn't finish the course and the final mark is set as *withdrawn*.

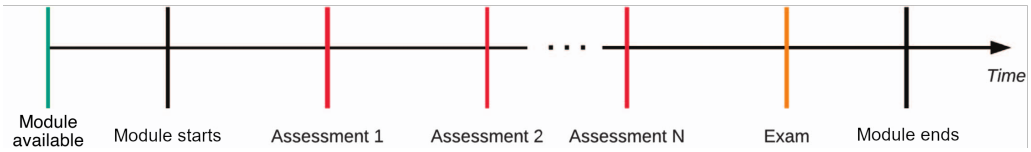


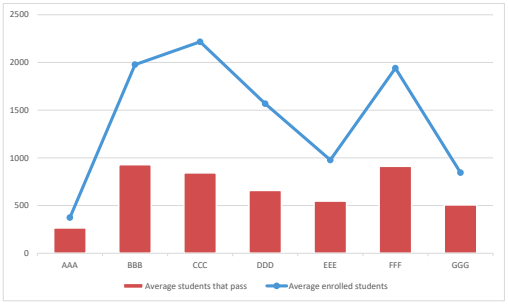
Figure 1. Typical course structured [13].

Table 2 shows a summary of available information for each course considering the number of times that the course has been offered, the average number of students per course and its standard deviation (considering the different times that it has been offered), the number of assignments by course, the average number of assignments submitted by students, and the percentage of students that fail or drop out the course relative to the total enrolled students. Assignments are divided between TMA and CMA types, as it has been commented previously.

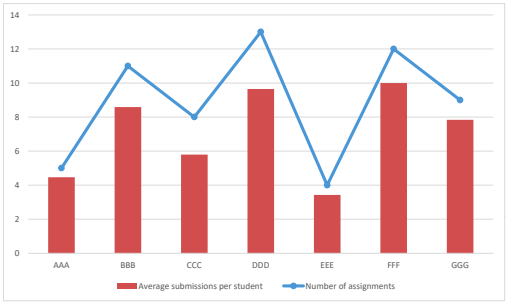
Table 2. Information of each course.

Course	Calls	Enrollments		Assignments		Submissions		No-Pass Rate
		Avg	SD	TMA	CMA	TMA	CMA	
AAA	2	374.00	12.73	5	0	4.47	–	29%
BBB	4	1977.25	338.34	6	5	4.47	4.12	53%
CCC	2	2217.00	397.39	4	4	2.89	2.91	62%
DDD	4	1568.00	354.18	6	7	4.63	5.02	58%
EEE	3	978.00	255.18	4	0	3.43	–	44%
FFF	4	1940.50	446.52	5	7	3.96	6.04	53%
GGG	3	844.67	102.00	3	6	2.69	5.15	40%

As we can see, there are significant differences between courses: they have been offered at different times during the considered academic years and the number of enrolled students also differs. There are also differences in terms of number and type of assignments, as well as the average number of submissions per student. Figure 2 shows the difference between courses. Figure 2a shows the average number of enrollments in a course versus the average pass rate. Figure 2b shows the number of assignments available by course versus the average submitted assignments per student. It can be observed that the number of assignments is different in each course and there are courses where the average percentage of submitted assignments is approximately 90% (as AAA course) while in other ones, as DDD course, this rate only reaches a 40% of submitted assignments. However, there is a tendency that seems to indicate that the more assignments are submitted, the more students pass the course. Thus, AAA course has a 71% of students that pass while DDD course has only a 42%.



(**a**) Enrollments by presentation



(**b**) Assignments by presentation

Figure 2. Information about enrolled students and submitted assignments.

4.2. Problem Representation Based on Assignment Information

In this study, prediction of student's performance to determine whether he/she will pass a course is focused on the information of submitted assignments. Table 3 shows the specific information provided by OULAD for assignments submitted by a student in each course: *assignment_type* is a categorical value specifying the two types of assignments considered (TMA and CMA). Each assignment has a weight (*assignment_weight*) and a score (*assignment_score*). Normally, the weighted sum of all assignments in each course is 100. The score is a numerical value in the range from 0 to 100. *Assessement_advanced* considers the number of days between the submission of the assignment by the student and its deadline. This is not a direct attribute in OULA dataset, but it can be calculated as the difference between the deadline date and the day on which student submitted it. Finally, it is considered *assignment_banked* that indicates if the assignment has been transferred from a previous enrollment in that course.

Table 3. Available information of each assignment.

Attribute	Description
assignment_type	Type of assignment: TMA or CMA.
assignment_weight	A number in range $[0, 100]$ that represents the weight of the assignment in the course.
assignment_advance	The number of days in advance with which the student submitted the assignment.
assignment_score	The score of the student in the assignment in range $[0, 100]$.
assignment_banked	A boolean flag that indicates if the assignment has been transferred from a previous presentation.

This study presents the traditional representation based on single instance and proposes a representation based on multiple instances learning to solve the problems of traditional representation. Since OULAD is presented in form of several CSV tables, it has been converted to ARFF format [49] using both mentioned representations. This process has implied the load of the dataset in a MySQL database and a slight restructuring of the data schema to ensure that it is maintained the Codd's normal form and data are not duplicated. Finally, from the database and through automated scripts, the different ARFF files with relationships considered have been generated. These datasets have been published in open access mode in the web repository associated with this paper (http://www.uco.es/kdis/mildistanceeducation/, accessed on 23 October 2021). Thus, a reproducible experimentation is facilitated to allow new advances in the area. The following sections define the representations proposed to solve the problem.

4.2.1. Representation Based on Single Instance Learning

As it has been commented, each student can submit a different number of assignments. Actually, assignments are not necessary to pass the course, although they are recommended to get a better understanding of the course. This information should be kept in an appropriate way so that it can influence in the prediction of student's academic achievement. That is, with the aim of predicting whether a student passes (with or without distinction) or does not pass (aggregating the failure and the dropout) a course.

The traditional supervised learning representation, used in previous studies with this dataset, is characterized by representing each student enrolled in a course during a semester as a pattern or vector of characteristics. Each pattern keeps the student's activity by means of a fixed number of attributes. According to the information specified in the Table 3, each assignment is represented by five attributes. Thus, each student is an pattern composed of $5 \times X$ attributes, being X the total of programmed assignments during that course and the final mark (student passes or fails the course). Moreover, the student's participation in

a course is specified by means of his/her identification, the course identification and the presentation identification that represents the edition of the course.

An illustrative example of problem representation can be seen in Figure 3. Here, we can see two students who belong to course AAA. Course AAA has 5 assignments. Therefore, it is necessary 25 attributes ($5 \times 5 = 25$) to represent information about student's assignments. One student submitted only two of the five assignments while the other one submitted all of them. As we can see, in traditional supervised learning, both students have the same number of attributes. Thus, if a student doesn't submit an assignment, the attributes related to this submission will have an empty value, but they have to be presented. This representation forces you to fill all attributes related to the non-submitted assignments, so there is a potential increase of the computational and storage resources needed for courses with a representative number of assignments.

student	course	presentation	1_asmnt_type	1_asmnt_weight	1_asmnt_advance	1_asmnt_score	1_asmnt_banked	2_asmnt_type	2_asmnt_weight	2_asmnt_advance	2_asmnt_score	2_asmnt_banked	3_asmnt_type	3_asmnt_weight	3_asmnt_advance	3_asmnt_score	3_asmnt_banked	4_asmnt_type	4_asmnt_weight	4_asmnt_advance	4_asmnt_score	4_asmnt_banked	5_asmnt_type	5_asmnt_weight	5_asmnt_advance	5_asmnt_score	5_asmnt_banked	final result
107	AAA	2014B	TMA	10	1	73	0	TMA	20				TMA	20				TMA	20				TMA	30	1	30	0	no pass
381	AAA	2014J	TMA	10	0	72	0	TMA	20	0	85	0	TMA	20	0	83	0	TMA	20	-9	80	0	TMA	30	0	70	0	pass

Figure 3. Representation example based on simple instance of two students in course AAA.

The equivalent representation of the commented example in ARFF format to be processed by machine learning algorithms using Weka framework can be seen in Table 4. In this case, it is necessary to define the 25 attributes in the header of the file. The instances are defined one per line following the `@data` label. Each instance represents one student and each attribute is separated by comma in the same order that were defined in the header. Thus, although a student does not submit an assignment, the information related to that assignment has to be filled in the instance. Other problem in this approach is that representation depends on the course. Thus, whether the previous example of AAA course is compared with an example of DDD course shown in Table 5, as DDD course has 13 assignments instead of 5, the dataset would have 65 attributes instead of 25 attributes. As we can see, the representation becomes more inefficient in cases of students that submit a low number of assignments. Moreover, there is a limitation of working with different courses because the representation is not uniform between courses. It depends of the assignments by course.

Table 4. Fragment of ARFF header for simple instance representation in course AAA.

Code	Attributes
`@relation assignments-course-AAA`	$5 \times$ `assignment_type`
	$5 \times$ `assignment_weight`
`@attribute 1-assignment_type { TMA, CMA }`	$5 \times$ `assignment_advance`
`@attribute 1-assignment_weight numeric`	$5 \times$ `assignment_score`
`...`	$5 \times$ `assignment_banked`
	Total: 25 attributes
`@attribute 5-assignment_score numeric`	
`@attribute 5-assignment_banked numeric`	
`@attribute final_result { pass, no_pass }`	
`@data`	

4.2.2. Representation Based on Multiple Instance Learning

MIL allows a flexible representation that adapts itself to the specific information available for each student according to his/her work in the course. In MIL representation,

each pattern is called bag and represents a student enrolled in a course during one semester. Each bag represents the student's activity. Thus, the bag is composed of a variable number of instances being each instance an assignment submitted by the student. Therefore, each bag has as many instances as assignments submitted by the student during a presentation of a course and one class attribute that can take, similarly to traditional representation, two values: the student passes (with distinction or without it) or does not pass the course (aggregating the failure and the dropout). This representation fits the problem perfectly because it can be customized by each student. Thus, the number of attributes in an instance is always the same, while the number of instances in a bag depends on the student's activity. There are five attributes in every instance described in Table 3: type, weight, days between the submission and the deadline, score obtained by the student and a status flag that indicates if the given assignment has been transferred from a previous presentation coursed by the student.

Table 5. Fragment of ARFF representation for simple instance representation in course DDD.

Code	Attributes
`@relation assignments-course-DDD`	13 × `assignment_type`
	13 × `assignment_weight`
`@attribute 1-assignment_type { TMA, CMA }`	13 × `assignment_advance`
`@attribute 1-assignment_weight numeric`	13 × `assignment_score`
`...`	13 × `assignment_banked`
	Total: 65 attributes
`@attribute 7-assignment_type { TMA, CMA }`	
`@attribute 7-assignment_weight numeric`	
`...`	
`@attribute 13-assignment_score numeric`	
`@attribute 13-assignment_banked numeric`	
`@attribute final_result { pass, no_pass }`	
`@data`	

The same example presented in traditional supervised learning (see previous Section 4.2.1) is addressed in Figure 4 from a flexible representation based on MIL. There are two students enrolled on AAA course: one of them submits only two assignments and he/she doesn't pass the course while the other one submits all assignments and he/she passes the course. In case of MIL, the data representation is much more efficient: each student is represented as a bag with so many instances as assignments he/she had submitted. As we can see, with this representation there are no empty fields and the representation. It is adapted perfectly to the available information of each student. The corresponding ARFF representation used by machine learning algorithms in Weka can be seen in Table 6. In this case, attributes for each instance must be defined as part of a *relational* attribute. Thus, they do not depend on the number of assignments in the course achieving a uniform representation between courses. For all course there are one relational attribute (with five instance attributes) independently of the number of assignments by course. Thus, the ARFF representation for DDD course would use the same number of attributes than AAA course. Each student is represented by one bag with all their instances enclosed in double quotes and each one separated by the character "`\n`" representing each submitted assignments.

Table 6. Fragment of ARFF header for multiple instance representation in any course.

Code	Attributes
`@relation assignments-course`	1 × `assignment_type` 1 × `assignment_weight`
`@attribute id_student-code_course-code_presentation {...}`	1 × `assignment_advance`
`@attribute bag relational`	1 × `assignment_score`
`@attribute assignment_type { TMA, CMA }`	1 × `assignment_banked`
`@attribute assignment_weight numeric`	Total: 5 attributes
`@attribute assignment_advance numeric`	
`@attribute assignment_score numeric`	
`@attribute assignment_banked numeric`	
`@end bag`	
`@attribute final_result { pass, no_pass }`	
`@data`	

student	course	presentation	asmnt_type	asmnt_weight	asmnt_advance	asmnt_score	asmnt_banked	final_result
107	AAA	2014B	TMA	10	1	73	0	no pass
			TMA	30	1	30	0	
381	AAA	2014J	TMA	10	0	72	0	pass
			TMA	20	0	85	0	
			TMA	20	0	83	0	
			TMA	20	-9	80	0	
			TMA	30	0	70	0	

Figure 4. MIL representation example of two students in course AAA.

5. Experimentation and Results

The goal of the experimental study is to investigate the potential of assignments to predict whether a student will or won't pass a course. As it has been commented in related work, the previous studies focus on this problem with OULA dataset involve mainly the evaluation of the student interactions with resources in VLE to determine his/her success in the course. On the contrary, this paper explores the potential of assignments to determine the level of engagement of students in a particular course. To validate this hypothesis, the performance of same algorithms will be analyzed. Thus, same information is used, but it is represented in one case from a traditional approach and in another case from a MIL approach. Thus, the flow of the experimental study is divided in five steps: first the configuration of the algorithms used to predict the student performance is presented in Section 5.1. Secondly, in Section 5.2 two procedures that permit to algorithms perform with MIL problems are presented and configured. Then, Section 5.3 defines the evaluation metrics as well as their meaning from a classification perspective and from a educational perspective. Next, Section 5.4 addresses the results contextualizing them in two comparative studies: attending to representations and attending to previous works. Finally, in Section 5.5, a discussion of the obtained results is carried out.

5.1. Configuration of Classification Algorithms

The experimentation of this is designed to offer a fair comparison between MIL and the traditional single-instances paradigm evaluating the same metrics in the same problem with the same information and in the same wide set of algorithms, part of the state of the art in supervised learning. They have been selected 23 algorithms considering the

main paradigms of machine learning and the most popular methods for predicting student performance (see Section 2.2).

The experimentation has been developed using Weka [49], a framework for machine learning in Java. In order to ensure solid evaluation, each experiment is executed with a 10-fold cross-validation. In addition, stochastic algorithms are executed 5 times with different seeds, having a total of 50 executions per algorithm and course. The datasets in ARFF format ready to be used in Weka have been published in open access mode in the web repository associated to this paper (http://www.uco.es/kdis/mildistanceeducation/, accessed on 23 October 2021). In order to easily reproduce the experimentation, this section presents the studied algorithms as well as their configurations. Since the purpose of this study is to compare types of learning under equal conditions, the configuration of the predictive algorithms should not favor one or the other representation paradigm. Thus, these configurations has been chosen based on the default settings that the authors specified according to the Weka workbench [49], where more information can be consulted.

The 23 predictive algorithms and their configurations are listed bellow:

- Methods based on trees : Decision Stump [50], J48 [51], Random Tree [52] and Random Forest [53]. See configurations in Table 7.
- Methods based on rules: ZeroR [49], OneR [54], NNge [55], PART [56] and Ridor [57]. See configurations in Table 7.
- Method based on Bayesian models: naive Bayes classifier [58]. See configuration in Table 7.
- Methods based on logistic regression: the algorithm considered in this paradigm is the proposal of Cessie and Houwelingen [59]. See configuration in Table 7.
- Methods based on Support Vector Machines (SVM): LibSVM [60], SGD with SVM as loss function [61], SMO with polynomial kernel [62] and SPegasos [63]. See configurations in Table 8.
- Methods based on Artificial Neural Networks (ANN): Multilayer Perceptron [64] and RBFNetwork [64]. See configurations in Table 8.
- Methods based on ensembles: AdaBoostM1 [65] and Bagging [66]. These meta-algorithms have been used with three distinct base classifiers previously commented: Random Forest, PART and Naive Bayes. See configurations in Table 8.

5.2. Configuration of Wrappers for MIL

With respect to MIL representation based on multiple instance, it is proposed the use of two different wrappers available in Weka [49] to adapt the MIL problem to single instance or traditional learning problem. Once that the problem is transformed, the same algorithms used in single instance representation (presented in previous Section 5.1) can be used with MIL representation. Thus, it is a more fair comparison because same algorithms and configurations are used. The proposals of MIL wrappers are the following:

- SimpleMI [67]: this wrapper makes a summary of all the instances of a bag in order to build a unique instance that can be processed by a simple instance algorithm.
- MIWrapper [68]: this wrapper assumes that all instances contribute equally and independently to the bag's label. Thus, the method breaks up the bag into its individual instances labeling each one with the bag label and assigning weights proportional to the number of instances in a bag. At evaluation time, the final class of the bag is derived from the classes assigned to its instances.

In the case of SimpleMI, there are two possible configurations to compute the summary of the instances of a bag into a single instance:

- Configuration 1: computing arithmetic mean of each attribute using all instances of the bag and using it in the summarized instance.
- Configuration 2: computing geometric mean of each attribute using all instances of the bag and using it in the summarized instance.

Table 7. Configuration of the studied algorithms I.

Algorithm	Parameter	Value	Algorithm	Parameter	Value
Decision Stump	-	-	**ZeroR**	-	-
J48	binarySplits	False	OneR	minBucketSize	6
	collapseTree	True	NNge	numAttemptsOf GeneOption	5
	confidenceFactor	0.25		numFolderMIOption	5
	doNotMakeSplit PointActualValue	False	PART	binarySplits	False
	minNumObj	2		confidenceFactor	0.25
	numFolds	3		doNotMakeSplit PointActualValue	False
	reduceErrorPruning	False		minNumObj	2
	useLaplace	False		numFolds 3	
	useMDLcorrection	True		reduceErrorPruning	False
Random Tree	allowUnclassified Instances	False		useMDLcorrection	True
	breakTiesRandomly	False	Ridor	folds	3
	maxDepth	0		majorityClass	False
	minNum	1.0		minNo	2.0
	minVarianceProp	0.001		shuffle	1
Random Forest	bagSizePercent	100		wholeDataErr	False
	breakTiesRandomly	False	Naive Bayes	useKernelEstimator	False
	computeAttribute Importance	False		useSupervised Discretization	False
	maxDepth	0	Logistic	maxIts	−1
	numFeatures	0		ridge	1×10^{-8}
	numIterations	100		useConjugate GradientDescent	False

In the case of MIWrapper, three configurations to compute the final class of the bag, extracted from the classes assigned at evaluation time:

- Configuration 1: computing the arithmetic average of the class probabilities of all the individual instances of the bag.
- Configuration 2: computing the geometric average of the class probabilities of all the individual instances of the bag.
- Configuration 3: checking the maximum probability of single positive instances. If there is at least one instance with its positive probability greater than 0.5, the entire bag is positive.

This study evaluates the accuracy of the different configurations to predict if a student will pass or fail the course. The experimentation consists of a 10-fold stratified cross-validation for every combination of wrapper configuration, algorithm and course. The complete results of this experimentation can be downloaded from the web repository associated to this work (http://www.uco.es/kdis/mildistanceeducation/, accessed on 23 October 2021).

With the average accuracy of the cross-validation, a statistical analysis is carried out in order to find significant differences between configurations in each MIL wrapper. Concretely, it is used the non-parametric Wilcoxon signed-rank test [69] to carry out a pairwise statistical procedure between every two configurations. In each comparison is applied the test and obtained a p-value independent to show if algorithms obtain significantly better accuracy values with a specific configuration. Table 9 shows the R^+, R^- and p-values for all pairwise comparisons carried out. For both wrappers and considering a confidence level of 99%, the configuration 1 obtains significantly higher accuracy values than the others. Thus, for SimpleMI is more convenient to summarize the bag with the

arithmetic mean and in case of MIWrapper, it is also better to use the arithmetic mean to combine the class probabilities of instances into the final class bag.

Table 8. Configuration of the studied algorithms II.

Algorithm	Parameter	Value	Algorithm	Parameter	Value
LibSVM	SVMType	C-SVC	Multilayer Perceptron	decay	False
	coef0	0		hiddenLayers	a
	cost	1.0		learningRate	0.3
	degree	3		momentum	0.2
	doNotReplace MissingValues	False		normalize Attributes	True
	eps	0.001		reset	True
	gamma	0.0		trainingTime	500
	kernelType	radial		validation Threshold	20
	normalize	False	RBF Network	maxIts	-1
	probability Estimates	False		minStdDev	0.1
	shrinking	True		numClusters	2
SGD	dontNormalize	False		ridge	1×10^{-8}
	dontReplace Missing	False	AdaBoost-Random Forest	numIterations	10
	epochs	500		useResampling	False
	lambda	1×10^{-4}		weightThreshold	100
	learningRate	0.01	AdaBoost-PART	numIterations	10
	lossFunction	SVM		useResampling	False
SMO	buildCalibration Models	False		weightThreshold	100
	c	1.0	AdaBoost-Naive Bayes	numIterations	10
	epsilon	1×10^{-12}		useResampling	False
	filterType	Normalize training		weightThreshold	100
	kernel	PolyKernel	Bagging—Random Forest	bagSizePercent	100
	tolerance Parameter	0.001		numIterations	10
SPegasos	dontNormalize	False	Bagging—PART	bagSizePercent	100
	dontReplace Missing	False		numIterations	10
	epochs	500	Bagging—Naive Bayes	bagSizePercent	100
	lambda	1×10^{-4}		numIterations	10
	lossFunction	SVM			

Table 9. Wilcoxon signed-rank test between MIL wrappers.

Wrapper	Comparison	R^+	R^-	p-Value
SimpleMI	**Conf. 1** vs. Conf. 2	12,740.5	300.5	4.78×10^{-25}
MIWrapper	**Conf. 1** vs. Conf. 2	8274.0	4606.0	5.19×10^{-4}
	Conf. 1 vs. Conf. 3	12,846.0	195.0	2.32×10^{-25}
	Conf. 2 vs. Conf. 3	12,847.0	194.0	2.28×10^{-25}

5.3. Evaluation Metrics

The metrics used for evaluation are some of the most common ones in the field of classification. In this context, classical concepts of binary classification are re-defined to our specific problem of having success in a course (passing it with or without distinction) or not having it (failure or dropout) as follow:

- t_p is the number of students correctly identified to pass the course.
- t_n is the number of students correctly identified to fail the course.
- f_p is the number of students do not correctly identified to pass the course (it is predicted that students pass the course, but they really do not pass).
- f_n is the number of students do not correctly identified to fail the course (it is predicted that students do pass the course, but they really pass).

Given the nature of the problem, in this context it is specially interesting to focus on students who are likely to fail. Thus, the metrics studied are [70]:

- *Accuracy* is the proportion of correctly classified students, i.e., identifying if they pass or not the course.

$$Acc = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \quad (1)$$

- *Sensitivity* is the proportion of students correctly classified that pass the course.

$$Se = \frac{t_p}{t_p + f_n} \quad (2)$$

- *Specificity* is the proportion of students correctly classified that do not pass the course.

$$Sp = \frac{t_n}{t_n + f_p} \quad (3)$$

5.4. Comparative Study

This section presents experimental results in the problem of comparing both multiple instances and single instance representation in predicting student performance using only his/her assignments activity. First, it is evaluated the performance of 23 machine learning algorithms using both representations. Statistical tests are used to determine if there are significant differences between performance using the different representations. Then, the best results achieved in this study are compared with the results of previous works shown in Section 3 that also predict the success of students for the same public dataset but using other student information available in OULAD.

5.4.1. Comparative Analysis between Different Representations

This section compares the performance of a wide set of algorithms in the problem of predicting student's success in a distance course using the same student information with different representations: traditional representation (based on single instance learning) and flexible representation (based on MIL). For solving the problem using flexible representation, as it is commented in Section 5.2, two different methods (MIWrapper and SimpleMI) that transform the MIL problem are used.

The experimental study carries out a 10-fold stratified cross-validation. The full results of this experimentation can been downloaded from the web repository associated to this work (http://www.uco.es/kdis/mildistanceeducation/, accessed on 23 October 2021).

Tables 10 and 11 show average accuracy results for each course. Thus, for each representation and algorithm, it is presented the average accuracy results of each course presented in OULAD. It can be observed that SimpleMI (using flexible representation) obtains the highest accuracy for most algorithms in the different courses. Thus, with an accuracy between 85% and 95% for all algorithms, SimpleMI outperforms in a robust way to traditional representation. MIWrapper (also using flexible representation) achieves similar results to SimpleMI and it obtains better results for the most algorithms than traditional representation. Although its values are somewhat lower. This affects to general accuracy of MIWrapper (around 80%) that is lower than those of SimpleMI. Algorithms that use traditional representation have a more variable performance. It can be appreciated that in general, this representation obtains lower accuracy (around 65%). In this case, we appreciate that more complex algorithms like the multi-layer perceptron, LibSVM or Ridor are needed to reach results comparable to SimpleMI. This is a disadvantage in terms of interpretability, since these methods do not provide information of which are the most relevant attributes in order to obtain representative information to help students. In this line, methods like those based on rules or trees get to outperform their results using SimpleMI representation while maintaining interpretable results. Concretely, they

obtain a 20% more than accuracy in average, using the same data but with a more optimal representation that fixes better to the problem.

For a more detailed analysis, Table 12 shows the average results for accuracy, sensitivity and specificity considering average results of the seven courses by the different algorithms. Results are grouped by representation: traditional representation and flexible representation (SimpleMI and MIWrapper). A full report of the results can be seen at the web repository associated to the article (http://www.uco.es/kdis/mildistanceeducation/, accessed on 23 October 2021). These data help to see in more detail tendencies like the superiority of SimpleMI, that gets the best accuracy results in all courses. Thus, the general tendency is that flexible representation (using SimpleMI) gets to improve the algorithms performance, obtaining better accuracy values versus traditional representation. In addition, this table shows in-depth the differences of performance between methods with the different representations. On the one hand, it is shown that traditional representation causes that algorithms obtain better values for specificity (predicting students that do not pass the course) at the expense of obtaining worse values for sensitivity (predicting students that pass the course). On the other hand, flexible representation (using WrapperMI) entails that algorithms obtain better values for sensitivity (predicting students that pass the course) at the expense of obtaining worse values for specificity (predicting students that do not pass the course). The fact of this off-balance between these measures is traduced in worse predictions overall. Again, flexible representation (using SimpleMI) obtains the most balanced results for both measures, sensitivity and specificity. Thus, this representation gets the best value or a very close one, achieving the best accuracy.

To analyze final results and show if there is significant differences between the behavior of algorithms using different representations, it is applied the test of Wilcoxon signed-ranks [69]. Thus, a pairwise comparison is carried out facing representation based on single instance (traditional) and the two MIL-based representations (SimpleMI, MIWrapper). Table 13 shows the results of the tests attending to accuracy measure. It is shown the R^+, R^- and p-values. With a confidence level of 99%, SimpleMI shows an improvement over the other representations. With a confidence level of 95%, MIWrapper shows an improvement over traditional representation.

Analyzing the differences in sensitivity, Table 14 shows a similar tendency: both flexible representations significantly outperform with a confidence level of 99% to traditional representation. However, attending to specificity results in Table 15, it can be appreciated that MIWrapper has difficulties to distinguish the negative class, which leads to a bad performance compared to SimpleMI and traditional representation. However, SimpleMI does not have this problem, reaching the best results in this metric too.

Table 10. Accuracy results by algorithm and courses I/II.

		AAA			BBB			CCC		
		Traditional	**SimpleMI**	**WrapperMI**	**Traditional**	**SimpleMI**	**WrapperMI**	**Traditional**	**SimpleMI**	**WrapperMI**
Trees methods	DecisionStump	0.710	**0.918**	0.789	0.655	**0.823**	0.627	**0.794**	0.788	0.740
	J48	0.705	**0.919**	0.809	0.569	**0.901**	0.770	0.622	**0.899**	0.845
	RandomTree	0.693	**0.868**	0.796	0.624	**0.853**	0.750	0.701	**0.860**	0.832
	RandomForest	0.727	**0.904**	0.818	0.619	**0.886**	0.754	0.691	**0.887**	0.843
Rules methods	ZeroR	0.710	**0.752**	**0.752**	0.525	**0.618**	**0.618**	**0.622**	0.509	0.509
	OneR	0.710	**0.918**	0.775	0.653	**0.902**	0.627	0.800	**0.889**	0.712
	NNge	**0.909**	0.898	0.779	0.501	**0.877**	0.657	0.415	**0.868**	0.578
	PART	0.709	**0.920**	0.812	0.569	**0.899**	0.779	0.630	**0.900**	0.846
	Ridor	0.899	**0.902**	0.744	**0.915**	0.859	0.569	**0.901**	0.850	0.655
	NaiveBayes	0.730	**0.921**	0.810	**0.785**	0.735	0.706	**0.823**	0.779	0.807
	Logistic	0.722	**0.925**	0.788	0.797	**0.807**	0.740	**0.904**	0.837	0.822
SVM methods	LibSVM	0.806	**0.916**	0.759	**0.851**	0.832	0.658	0.830	**0.861**	0.752
	SPegasos	0.301	**0.913**	0.759	0.489	**0.795**	0.627	0.454	**0.824**	0.662
	SGD	0.733	**0.915**	0.755	0.755	**0.805**	0.627	0.815	**0.836**	0.729
	SMO	0.721	**0.918**	0.759	0.741	**0.808**	0.735	0.806	**0.837**	0.815
ANN methods	RBFNetwork	0.758	**0.859**	0.813	0.856	**0.907**	0.687	**0.882**	0.873	0.829
	MultilayerPerceptron	0.879	**0.919**	0.808	0.911	**0.939**	0.760	**0.918**	0.907	0.837
Ensembles methods	AdaBoost&RandomForest	0.656	**0.905**	0.813	0.743	**0.885**	0.754	0.689	**0.884**	0.838
	AdaBoost&PART	0.738	**0.918**	0.811	0.699	**0.894**	0.778	0.715	**0.894**	0.847
	AdaBoost&NaiveBayes	0.730	**0.921**	0.809	0.790	**0.805**	0.753	**0.836**	0.788	0.830
	Bagging&RandomForest	0.726	**0.919**	0.820	0.625	**0.905**	0.760	0.690	**0.903**	0.845
	Bagging&PART	0.715	**0.915**	0.811	0.562	**0.895**	0.778	0.629	**0.892**	0.848
	Bagging&NaiveBayes	0.732	**0.920**	0.809	**0.785**	0.699	0.683	**0.826**	0.777	0.795

Table 11. Accuracy results by algorithm and courses II/II.

		DDD			EEE			FFF			GGG		
		Traditional	**SimpleMI**	**WrapperMI**	**Traditional**	**SimpleMI**	**WrapperMI**	**Traditional**	**SimpleMI**	**WrapperMI**	**Traditional**	**SimpleMI**	**WrapperMI**
Trees methods	DecisionStump	0.667	**0.775**	0.724	0.646	**0.903**	0.718	0.661	**0.906**	0.797	0.607	**0.907**	0.717
	J48	0.584	**0.863**	0.809	0.593	**0.898**	0.816	0.530	**0.931**	0.868	0.597	**0.908**	0.784
	RandomTree	0.584	**0.828**	0.776	0.596	**0.872**	0.789	0.667	**0.909**	0.830	0.590	**0.862**	0.771
	RandomForest	0.584	**0.855**	0.797	0.603	**0.888**	0.799	0.659	**0.929**	0.848	0.600	**0.900**	0.771
Rules methods	ZeroR	**0.584**	0.528	0.528	0.562	**0.718**	**0.718**	0.530	**0.580**	**0.580**	0.598	**0.717**	**0.717**
	OneR	0.665	**0.844**	0.668	0.646	**0.904**	0.729	0.656	**0.924**	0.611	0.603	**0.907**	0.720
	NNge	**0.863**	0.830	0.695	**0.911**	0.871	0.722	0.506	**0.918**	0.637	0.612	**0.887**	0.635
	PART	0.584	**0.862**	0.812	0.588	**0.900**	0.805	0.578	**0.931**	0.874	0.593	**0.908**	0.791
	Ridor	**0.876**	0.833	0.618	**0.907**	0.892	0.739	**0.941**	0.915	0.670	**0.905**	0.868	0.697
	NaiveBayes	**0.817**	0.765	0.742	0.602	**0.900**	0.773	0.789	**0.910**	0.832	0.588	**0.902**	0.814
	Logistic	0.833	**0.842**	0.780	0.802	**0.899**	0.794	**0.917**	0.915	0.826	0.610	**0.907**	0.743
SVM methods	LibSVM	0.814	**0.843**	0.724	0.884	**0.896**	0.761	0.845	**0.909**	0.662	**0.854**	0.846	0.7232
	SPegasos	0.583	**0.840**	0.767	0.443	**0.902**	0.748	0.495	**0.911**	0.601	0.598	**0.907**	0.717
	SGD	0.796	**0.847**	0.688	0.607	**0.895**	0.722	0.719	**0.911**	0.594	0.604	**0.907**	0.717
	SMO	0.787	**0.845**	0.771	0.592	**0.895**	0.781	0.786	**0.906**	0.793	0.600	**0.907**	0.717
	RBFNetwork	0.758	**0.799**	0.768	0.856	**0.902**	0.793	0.882	**0.909**	0.837	0.694	**0.904**	0.790
ANN methods	MultilayerPerceptron	**0.879**	0.855	0.787	**0.911**	0.903	0.807	0.918	**0.930**	0.870	0.881	**0.907**	0.784
Ensembles methods	AdaBoost&RandomForest	0.761	**0.857**	0.786	0.679	**0.889**	0.792	0.805	**0.926**	0.838	0.728	**0.898**	0.769
	AdaBoost&PART	0.623	**0.859**	0.808	0.721	**0.894**	0.799	0.701	**0.929**	0.873	0.677	**0.904**	0.793
	AdaBoost&NaiveBayes	**0.820**	0.814	0.767	0.818	**0.900**	0.779	0.838	**0.910**	0.855	0.588	**0.902**	0.809
	Bagging&RandomForest	0.584	**0.868**	0.798	0.602	**0.901**	0.803	0.667	**0.935**	0.854	0.602	**0.912**	0.775
	Bagging&PART	0.584	**0.861**	0.817	0.597	**0.896**	0.812	0.585	**0.931**	0.876	0.598	**0.905**	0.790
	Bagging&NaiveBayes	**0.818**	0.743	0.730	0.601	**0.900**	0.771	0.789	**0.911**	0.820	0.593	**0.902**	0.816

Table 12. Average results by algorithm for all courses.

		Accuracy			Sensitivity			Specificity		
		Traditional	**SimpleMI**	**WrapperMI**	**Traditional**	**SimpleMI**	**WrapperMI**	**Traditional**	**SimpleMI**	**WrapperMI**
Trees methods	DecisionStump	0.677	**0.860**	0.730	0.458	**0.949**	0935	**0.822**	0.703	0.300
	J48	0.600	**0.903**	0.814	0.434	0.953	**0.959**	0.592	**0.802**	0.505
	RandomTree	0636	**0.865**	0.792	0.504	0.889	**0.951**	0.616	**0.807**	0.463
	RandomForest	0640	**0.893**	0.804	0.510	0.931	**0.957**	0.610	**0.810**	0.485
Rules methods	ZeroR	0.590	**0.631**	**0.631**	0.429	**0.857**	**0.857**	**0.571**	0.143	0.143
	OneR	0.676	**0.898**	0.692	0.458	0.966	**0.985**	**0.818**	0.770	0.156
	NNge	0.674	**0.878**	0.672	**0.954**	0.905	0.891	0.397	**0.814**	0.298
	PART	0.607	**0.903**	0.817	0.443	0.949	**0.957**	0.604	**0.806**	0.515
	Ridor	**0.906**	0.874	0.670	0.948	**0.949**	0.871	**0.732**	0.720	0.291
	NaiveBayes	0.733	**0.844**	0.783	0.851	0.938	**0.955**	0.560	**0.676**	0.440
	Logistic	0.798	**0.876**	0.785	0.822	0.945	**0.962**	0.716	**0.738**	0.417
SVM methods	LibSVM	0.841	**0.872**	0.720	0.978	0.920	**0.989**	0.667	**0.769**	0.218
	SPegasos	0.480	**0.870**	0.697	0.548	0.948	**0.964**	0.483	**0.717**	0.193
	SGD	0.718	**0.874**	0.690	0.728	0.950	**0.968**	0.587	**0.720**	0.142
	SMO	0.719	**0.874**	0.767	0.722	0.953	**0.965**	0.585	**0.716**	0.354
ANN methods	RBFNetwork	0.812	**0.879**	0.788	0.861	0.947	**0.954**	0.647	**0.727**	0.433
	MultilayerPerceptron	0.899	**0.909**	0.808	0.919	0.943	**0.954**	0.757	**0.788**	0.498
Ensembles methods	AdaBoost&RandomForest	0.723	**0.892**	0.798	0.607	0.930	**0.953**	0.719	**0.810**	0.478
	AdaBoost&PART	0.696	**0.899**	0.816	0.684	0.946	**0.956**	0.617	**0.799**	0.514
	AdaBoost&NaiveBayes	0.774	**0.863**	0.800	0.843	0.925	**0.949**	0.617	**0.738**	0.488
	Bagging&RandomForest	0.642	**0.906**	0.808	0.513	**0.958**	0.957	0.610	**0.799**	0.495
	Bagging&PART	0.610	**0.899**	0.819	0.444	0.940	**0.959**	0.603	**0.811**	0.518
	Bagging&NaiveBayes	0.735	**0.836**	0.775	0.849	0.945	**0.958**	0.525	**0.648**	0.416

Table 13. Wilcoxon signed-rank test results for accuracy measure.

Comparison	R^+	R^-	*p*-Value
SimpleMI vs. Traditional	273	3	1.192×10^{-6}
MIWrapper vs. Traditional	212	62	0.02332
SimpleMI vs. MIWrapper	275	0	4.768×10^{-7}

Table 14. Wilcoxon signed-rank test results for sensitivity measure.

Comparison	R^+	R^-	*p*-Value
SimpleMI vs. Traditional	269	7	4.53×10^{-6}
MIWrapper vs. Traditional	269	7	4.53×10^{-6}
SimpleMI vs. **MIWrapper**	44	231	0.003252

Table 15. Wilcoxon signed-rank test results for specificity measure.

Comparison	R^+	R^-	*p*-Value
SimpleMI vs. Traditional	239	37	0.001279
MIWrapper vs. **Traditional**	0	276	2.384×10^{-7}
SimpleMI vs. MIWrapper	275	0	4.768×10^{-7}

The main conclusion extracted from this experimentation is the importance of an appropriate problem representation. It can be seen that the assignments represented with single instance learning obtain lower results. These results can explain why assignments are not widely included as influencing factors in previous studies of predicting students' performance. Using the same type of information and the same learning algorithms, but a representation based on MIL, algorithms can predict the student's success in distance education with more accuracy. Thus, we can see that flexible representation can obtain differences of more than 20% of performance in comparison with traditional representation.

5.4.2. Comparative Analysis with Previous Works

Based on previous studies shown in Section 3 (Table 1), this work deviates from the general trend marked by the use of clickstreams (well known as student interactions with the VLE). As it has been shown, a limited number of studies use the information about assignments as influential factor for the predictions.

However, as it has been shown in the previous comparative study, assignments can obtain equal or better results if they are processed with the appropriate representation. Thus, in this section a comparison of the accuracy to predict student's performance is carried out attending to the best MIL method according previous section, SimpleMI, and the related work. Table 16 shows these differences with a special focus on previous works that have used same algorithms but from a traditional learning perspective. Thus, it is shown the algorithm and the data from OULAD in previous work compared to the use of MIL. For example, we can focus on the unique previous work that uses solely assignments data [40]. This work is limited to courses CCC and FFF and it obtains an average accuracy of 83% using decision trees. In our work, for these courses and algorithms, the best results reached show an average accuracy of 92'1%. Focusing on previous works focus on predicting student success or failure in a course but not based on assignments data, the best results are achieved by [30]. This work applies J48 over VLE activity data obtaining an average accuracy over all the courses 90%. In our case, using only assignment data with the same algorithm, the best result achieved for each course gets an average accuracy of 92.7%.

Table 16. Comparison with previous works that uses same algorithms from a traditional learning perspective.

Algorithm	Course	Previous Work					Proposed MIL Approach			
		Ref.	Assign.	Clicks	Demog.	Acc.	Assign.	Clicks	Demog.	Acc.
Decision tree	CCC	[40]	X			86.6%	X			**89.9%**
	FFF					79.4%	X			**93.1%**
	AAA	[44]			X	83.1%	X			**91.9%**
J48	All courses	[30]		X		86.7%	X			**90.3%**
		[33]		X		88.5%				
RandomForest	All courses	[47]	X	X	X	81.8%	X			**89.3%**
		[26]		X		86.2%				
NaiveBayes	All courses	[38]		X	X	63.8%	X			**84.4%**
SVM	All courses	[27]		X	X	**88.0%**	X			87.2%
ANN	All courses	[45]		X		89.0%	X			**90.9%**

There are also some works that use algorithms not included in our study, like XGBoost [36], Gausian Mixture Models [29], or Deep Learning [39,41–43]. However, if their best results are compared to the best results obtained in this work, MIL achieves equals or better results using less data and in a more interpretative way. This is shown in Table 17. This table shows each previous work that employs an algorithm not included in our study and it addresses the same problem of predicting student's performance, as well as the information used, with the best result obtained using MIL (SimpleMI) for the same courses.

Table 17. Comparison with previous works that use different algorithms from a traditional learning perspective.

Course	Previous Work						Proposed MIL Approach				
	Ref.	Algorithm	Assign.	Clicks	Demog.	Acc.	Algorithm	Assign.	Clicks	Demog.	Acc.
BBB	[29]	Gaussian Mixture		X		85.5%	Multilayer Perceptron	X			**93.9%**
DDD	[37]	Dynamic Incremental Semi-supervised Fuzzy C-means	X	X	X	**89.3%**	Bagging & RandomForest	X			86.8%
AAA	[43]	Convolutional and Recurrent Deep Model	X	X	X	61.0%	PART	X			**92.0%**
All courses	[41]	Deep ANN	X	X	X	84.5%	Multilayer Perceptron	X			**90.9%**
	[39]	Recurrent Neural Network		X	X	75.0%					
	[45]	Adversarial Network + ANN		X		89.0%					

Nevertheless, these comparisons should be taken carefully, since the conditions of experimentation and evaluation may differ. Thus, our study shows that general performance of predictive models based on MIL and assignments data are competitive with respect to other approaches that do not use this information or combine it with other factors. To promote future fair comparisons, all information necessary to carry out this experimental study is available in the public repository associated to the article.

5.5. Discussion of Results

In this section, it is highlighted the implications of using MIL to predict students' performance in a context of distance learning. Moreover, the possible limitations that the proposed approach may have are also discussed. Comparative analysis between different representations has shown that MIL robustly outperforms to traditional representation. Concretely, it is obtained approximately a 20% more accurate results on average. In the context of the problem of student's performance, it is important to attend not only to general metrics that give the same importance to all classes, but also to metrics that explain how well our system finds those problematic students that will not pass or dropout the course. This can be measured with metrics like specificity, in which MIL also outperforms

traditional data representation. Given the superiority in performance, it is also worth commenting on the advantage of using MIL in terms of interpretability of results. In the context of predicting student's performance, it is very important that the models are not a black box. Thus, they could be interpreted by mentors and tutors in order to be able to correct in time trends that may lead to students failing or dropping out of the course. With MIL, a student's assignment information takes up to four times less than using the simple instance-based representation. This helps to create models more quickly and reduce redundancy in the information, which makes the results easier for a human to read. This, together with the lack of the need for deep learning or black box algorithms to obtain results above 90% accuracy, means that the level of interpretability of MIL models remains high and can be used in real-world tools to identify potential problems in distance learning courses with large numbers of failures, as well as to identify specific students at risk of failing.

On the other side, the approach of this work may also have limitations that should be taken into account. The authors can identify two main problems that can be addressed in future works. First, this work has been carried out by analyzing the dropout and failure profiles together. Although both profiles correspond to students who do not pass the course, this could be due to different causes, so it is worth considering the possibility of carrying out a separate analysis to identify each type of student at risk. Second, the study has been made considering all the activities of the course, i.e., it is required to have reached an advanced point of the course to have all the information of the assignments that have been submitted. This limits the possibilities of action to prevent an at-risk student from failing the course. It would be desirable to tweak the approach to only use the information from the assignments up to a certain point in the course, in order to have enough time before the final exam to be able to adequately guide the at-risk student to avoid failure.

6. Conclusions and Future Work

This paper shows the impact of assignments information to predict the academic achievements. Online courses are characterized by a high number of enrolled students with a low participation and engagement, in general terms. Moreover, assignments depend on each course, because each has its own curriculum, scheduling and evaluation approach, so offers different number and type of assignments. This has led to ignore assignments as a criterion to predict students performance, as proves the very limited number of works that study them. The main problem is that traditional representation produces a very complex representation that machine learning algorithms cannot properly process.

This work shows that information about assignments can be very valuable to predict students performance when it is appropriately represented. The comparative study has employed a public dataset in learning analytics, OULAD. This dataset allows to work with a big amount of data and the comparison of the proposed study and results in an existing common framework. Thus, starting from this dataset, the appropriate transformations have been applied to use MIL as the learning paradigm, generating the files in ARFF format necessary to train the predictive models. These files are publicly available in the web repository associated with the article. Experimental results over a wide set of 23 machine learning algorithms and 7 courses show that, in a general way, using assignments in a flexible representation improve the accuracy with respect to use the same information in a traditional representation, achieving an important balance between sensitivity and specificity measures. Statistical tests confirm these results showing significant differences in every studied metric between multiple instances and single instance representations. Finally, it is carried out a comparison with previous studies that also use OULAD for predicting student performance from other factors, such as demographic information and students interactions with VLE, showing the relevance of assignments as a very influential factor to determine the student success or failure.

The great variety of information gathered in OULAD together with the promising results obtained open the door to continue this line of research. Thus, it may be tested

another source of information to predict the student success like the clicks activity in the VLE, the number of times that the student has done a course or including demographic data. In addition, it is propose to extend the study to algorithms unique to MIL paradigm, as well as explore different MI assumptions existing in the bibliography.

Author Contributions: Conceptualization, A.E., C.R. and A.Z.; methodology, A.E., C.R. and A.Z.; software, A.E.; validation, A.E. and A.Z.; formal analysis, C.R. and A.Z.; investigation, A.E., C.R. and A.Z.; writing—original draft preparation, A.E., C.R.; writing—review and editing, A.Z.; supervision, C.R. and A.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Spanish Ministry of Science and Technology [No. PID2020-115832GB-100].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Source data are available https://analyse.kmi.open.ac.uk/open_dataset, accessed on 23 October 2021, and dataset processed with arff format can be found at https://www.uco.es/kdis/mildistanceeducation, accessed on 23 October 2021.

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

References

1. Yunusa, A.A.; Umar, I.N. A scoping review of Critical Predictive Factors (CPFs) of satisfaction and perceived learning outcomes in E-learning environments. *Educ. Inf. Technol.* **2021**, *26*, 1223–1270. [CrossRef]
2. Romero, C.; Ventura, S. Data mining in education. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2013**, *3*, 12–27. [CrossRef]
3. Tomasevic, N.; Gvozdenovic, N.; Vranes, S. An overview and comparison of supervised data mining techniques for student exam performance prediction. *Comput. Educ.* **2020**, *143*, 103676. [CrossRef]
4. Gardner, J.; Brooks, C. Student success prediction in MOOCs. *User Model. User Adapt. Interact.* **2018**, *28*, 127–203. [CrossRef]
5. Panagiotakopoulos, T.; Kotsiantis, S.; Kostopoulos, G.; Iatrellis, O.; Kameas, A. Early Dropout Prediction in MOOCs through Supervised Learning and Hyperparameter Optimization. *Electronics* **2021**, *10*, 1701. [CrossRef]
6. Gong, J.W.; Liu, H.C.; You, X.Y.; Yin, L. An integrated multi-criteria decision making approach with linguistic hesitant fuzzy sets for E-learning website evaluation and selection. *Appl. Soft Comput.* **2021**, *102*, 107118. [CrossRef]
7. Yang, Q.; Lee, Y.C. The Critical Factors of Student Performance in MOOCs for Sustainable Education: A Case of Chinese Universities. *Sustainability* **2021**, *13*, 8089. [CrossRef]
8. Jaggars, S.S.; Xu, D. How do online course design features influence student performance? *Comput. Educ.* **2016**, *95*, 270–284. [CrossRef]
9. Muñoz-Merino, P.J.; González Novillo, R.; Delgado Kloos, C. Assessment of skills and adaptive learning for parametric exercises combining knowledge spaces and item response theory. *Appl. Soft Comput. J.* **2018**, *68*, 110–124. [CrossRef]
10. Birjali, M.; Beni-Hssane, A.; Erritali, M. A novel adaptive e-learning model based on Big Data by using competence-based knowledge and social learner activities. *Appl. Soft Comput. J.* **2018**, *69*, 14–32. [CrossRef]
11. Abu Saa, A.; Al-Emran, M.; Shaalan, K. Factors Affecting Students' Performance in Higher Education: A Systematic Review of Predictive Data Mining Techniques. *Technol. Knowl. Learn.* **2019**, *24*, 567–598. [CrossRef]
12. Dietterich, T.G.; Lathrop, R.H.; Lozano-Pérez, T. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.* **1997**, *89*, 31–71. [CrossRef]
13. Kuzilek, J.; Hlosta, M.; Zdrahal, Z. Data Descriptor: Open University Learning Analytics dataset. *Sci. Data* **2017**, *4*, 1–8. [CrossRef] [PubMed]
14. Carbonneau, M.A.; Cheplygina, V.; Granger, E.; Gagnon, G. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognit.* **2018**, *77*, 329–353. [CrossRef]
15. Sudharshan, P.J.; Petitjean, C.; Spanhol, F.; Oliveira, L.E.; Heutte, L.; Honeine, P. Multiple instance learning for histopathological breast cancer image classification. *Expert Syst. Appl.* **2019**, *117*, 103–111. [CrossRef]
16. Zafra, A.; Romero, C.; Ventura, S. Multiple instance learning for classifying students in learning management systems. *Expert Syst. Appl.* **2011**, *38*, 15020–15031. [CrossRef]
17. Kotsiantis, S.; Kanellopoulos, D.; Tampakas, V. Financial application of multi-instance learning: Two Greek case studies. *J. Converg. Inf. Technol.* **2010**, *5*, 5.
18. Foulds, J.; Frank, E. A review of multi-instance learning assumptions. *Knowl. Eng. Rev.* **2010**, *25*, 1–25. [CrossRef]
19. Alyahyan, E.; Düştegör, D. Predicting academic success in higher education: Literature review and best practices. *Int. J. Educ. Technol. High. Educ.* **2020**, *17*, 1–21. [CrossRef]
20. Hasan, R.; Palaniappan, S.; Mahmood, S.; Abbas, A.; Sarker, K.U.; Sattar, M.U. Predicting student performance in higher educational institutions using video learning analytics and data mining techniques. *Appl. Sci.* **2020**, *10*, 3894. [CrossRef]

21. 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]
22. Shelton, B.E.; Hung, J.L.; Lowenthal, P.R. Predicting student success by modeling student interaction in asynchronous online courses. *Distance Educ.* **2017**, *38*, 59–69. [CrossRef]
23. Coussement, K.; Phan, M.; De Caigny, A.; Benoit, D.F.; Raes, A. Predicting student dropout in subscription-based online learning environments: The beneficial impact of the logit leaf model. *Decis. Support Syst.* **2020**, *135*, 113325. [CrossRef]
24. Kostopoulos, G.; Lipitakis, A.D.; Kotsiantis, S.; Gravvanis, G. Predicting student performance in distance higher education using active learning. *Commun. Comput. Inf. Sci.* **2017**, *744*, 75–86.
25. Kostopoulos, G.; Kotsiantis, S.; Fazakis, N.; Koutsonikos, G.; Pierrakeas, C. A Semi-Supervised Regression Algorithm for Grade Prediction of Students in Distance Learning Courses. *Int. J. Artif. Intell. Tools* **2019**, *28*, 1940001. [CrossRef]
26. Haiyang, L.; Wang, Z.; Benachour, P.; Tubman, P. A time series classification method for behaviour-based dropout prediction. In Proceedings of the 18th IEEE International Conference on Advanced Learning Technologies, Mumbai, India, 9–13 July 2018; pp. 191–195.
27. Heuer, H.; Breiter, A. Student Success Prediction and the Trade-Off between Big Data and Data Minimization. In *Die 16. E-Learning Fachtagung Informatik*; Krömker, D., Schroeder, U., Eds.; Gesellschaft für Informatik e.V.: Bonn, Germany, 2018; pp. 219–230.
28. Doijode, V.; Singh, N. Predicting student success based on interaction with virtual learning environment. In Proceedings of the SouthEast SAS Users Group Conference, Bethesda, MD, USA, 16–18 October 2016; p. 10.
29. Alshabandar, R.; Hussain, A.; Keight, R.; Laws, A.; Baker, T. The Application of Gaussian Mixture Models for the Identification of At-Risk Learners in Massive Open Online Courses. In Proceedings of the IEEE Congress on Evolutionary Computation, Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
30. Netto Silveira, P.D.; Lube Dos Santos, O. A predictive model of academic failure or success for institutional and trace data. In Proceedings of the 14th Latin American Conference on Learning Technologies, San Jose Del Cabo, Mexico, 30 October–1 November 2019; pp. 162–165.
31. Netto Silveira, P.D.; Cury, D.; Menezes, C.; Dos Santos, O.L. Analysis of classifiers in a predictive model of academic success or failure for institutional and trace data. In Proceedings of the IEEE Frontiers in Education Conference, Covington, KY, USA, 16–19 October 2019; pp. 1–8.
32. Kuzilek, J.; Vaclavek, J.; Fuglik, V.; Zdrahal, Z. Student Drop-out Modelling Using Virtual Learning Environment Behaviour Data. In *European Conference on Technology Enhanced Learning*; Springer: Cham, Switzerland, 2018; pp. 166–171.
33. Hussain, M.; Zhu, W.; Zhang, W.; Abidi, S.M.R. Student Engagement Predictions in an e-Learning System and Their Impact on Student Course Assessment Scores. *Comput. Intell. Neurosci.* **2018**, *2018*, 6347186. [CrossRef]
34. Hassan, S.U.; Waheed, H.; Aljohani, N.R.; Ali, M.; Ventura, S.; Herrera, F. Virtual learning environment to predict withdrawal by leveraging deep learning. *Int. J. Intell. Syst.* **2019**, *34*, 1935–1952. [CrossRef]
35. Aljohani, N.R.; Fayoumi, A.; Hassan, S.U. Predicting at-risk students using clickstream data in the virtual learning environment. *Sustainability* **2019**, *11*, 7238. [CrossRef]
36. Hlosta, M.; Zdrahal, Z.; Zendulka, J. Ouroboros: Early identification of at-risk students without models based on legacy data. In Proceedings of the 7th International Learning Analytics & Knowledge Conference, Vancouver, BC, Canada, 13–17 March 2017; pp. 6–15.
37. Casalino, G.; Castellano, G.; Mencar, C. Incremental and Adaptive Fuzzy Clustering for Virtual Learning Environments Data Analysis. In Proceedings of the International Conference on Information Visualisation, Paris, France, 2–5 July 2019; pp. 382–387.
38. Azizah, E.N.; Pujianto, U.; Nugraha, E.; Darusalam. Comparative performance between C4.5 and Naive Bayes classifiers in predicting student academic performance in a Virtual Learning Environment. In Proceedings of the 4th International Conference on Education and Technology, Malang, Indonesia, 26–28 October 2018; pp. 18–22.
39. He, Y.; Chen, R.; Li, X.; Hao, C.; Liu, S.; Zhang, G.; Jiang, B. Online at-risk student identification using RNN-GRU joint neural networks. *Information* **2020**, *11*, 474. [CrossRef]
40. Ho, L.C.; Jin Shim, K. Data Mining Approach to the Identification of At-Risk Students. In Proceedings of the IEEE International Conference on Big Data, Seattle, WA, USA, 10–13 December 2018; pp. 5333–5335.
41. Waheed, H.; Hassan, S.U.; Aljohani, N.R.; Hardman, J.; Alelyani, S.; Nawaz, R. Predicting academic performance of students from VLE big data using deep learning models. *Comput. Hum. Behav.* **2020**, *104*, 106189. [CrossRef]
42. Qiao, C.; Hu, X. A Joint Neural Network Model for Combining Heterogeneous User Data Sources: An Example of At-Risk Student Prediction. *J. Assoc. Inf. Sci. Technol.* **2019**, *71*, 1192–1204. [CrossRef]
43. Song, X.; Li, J.; Sun, S.; Yin, H.; Dawson, P.; Doss, R.R.M. SEPN: A Sequential Engagement Based Academic Performance Prediction Model. *IEEE Intell. Syst.* **2021**, *36*, 46–53. [CrossRef]
44. Rizvi, S.; Rienties, B.; Khoja, S.A. The role of demographics in online learning; A decision tree based approach. *Comput. Educ.* **2019**, *137*, 32–47. [CrossRef]
45. Waheed, H.; Anas, M.; Hassan, S.U.; Aljohani, N.R.; Alelyani, S.; Edifor, E.E.; Nawaz, R. Balancing sequential data to predict students at-risk using adversarial networks. *Comput. Electr. Eng.* **2021**, *93*, 107274. [CrossRef]
46. Hlosta, M.; Zdrahal, Z.; Zendulka, J. Are we meeting a deadline? classification goal achievement in time in the presence of imbalanced data. *Knowl.-Based Syst.* **2018**, *160*, 278–295. [CrossRef]

47. Adnan, M.; Habib, A.; Ashraf, J.; Mussadiq, S.; Raza, A.A.; Abid, M.; Bashir, M.; Khan, S.U. Predicting at-Risk Students at Different Percentages of Course Length for Early Intervention Using Machine Learning Models. *IEEE Access* **2021**, *9*, 7519–7539. [CrossRef]
48. Zafra, A.; Romero, C.; Ventura, S. DRAL: A tool for discovering relevant e-activities for learners. *Knowl. Inf. Syst.* **2013**, *36*, 211–250. [CrossRef]
49. Witten, I.H.; Frank, E.; Hall, M.A.; Pal, C.J. *Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed.; Elsevier: Saint Louis, MO, USA, 2016; pp. 1–621.
50. Quinlan, J.R. Induction of Decision Trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
51. Salzberg, S.L. C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Mach. Learn.* **1994**, *16*, 235–240. [CrossRef]
52. Drmota, M. *Random Trees: An Interplay between Combinatorics and Probability*; Springer: New York, NY, USA, 2009; pp. 1–458.
53. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
54. Holte, R.C. Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. *Mach. Learn.* **1993**, *11*, 63–90. [CrossRef]
55. Martin, B. Instance-Based Learning: Nearest Neighbor with Generalization. Master's Thesis, University of Waikato, Hamilton, New Zealand, 1995.
56. Frank, E.; Witten, I.H. Generating accurate rule sets without global optimization. In Proceedings of the Fifteenth International Conference on Machine Learning, Hamilton, New Zealand, 24–27 July 1998; pp. 144–151.
57. Gaines, B.R.; Compton, P. Induction of ripple-down rules applied to modeling large databases. *J. Intell. Inf. Syst.* **1995**, *5*, 211–228. [CrossRef]
58. John, G.H.; Langley, P. Estimating Continuous Distributions in Bayesian Classifiers. In Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–20 August 2013; pp. 338–345.
59. Cessie, S.L.; Houwelingen, J.C.V. Ridge Estimators in Logistic Regression. *Appl. Stat.* **1992**, *41*, 191. [CrossRef]
60. Chang, C.C.; Lin, C.J. LIBSVM: A Library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 1–27. [CrossRef]
61. Zhang, T. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In Proceedings of the Twenty-First International Conference on Machine Learning, Banff, AB, Canada, 4–8 July 2004; pp. 919–926.
62. Meng, L.; Wu, Q.H. Fast training of Support Vector Machines using error-center-based optimization. *Int. J. Autom. Comput.* **2005**, *2*, 6–12. [CrossRef]
63. Shalev-Shwartz, S.; Singer, Y.; Srebro, N.; Cotter, A. Pegasos: Primal estimated sub-gradient solver for SVM. *Math. Program.* **2011**, *127*, 3–30. [CrossRef]
64. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*; Springer: New York, NY, USA, 2009; p. 744.
65. Freund, Y.; Schapire, R.E. Experiments with a New Boosting Algorithm. In Proceedings of the 13th International Conference on Machine Learning, Bari, Italy, 3–6 July 1996; pp. 148–156.
66. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
67. Dong, L. A Comparison of Multi-Instance Learning Algorithms. Master's Thesis, University of Waikato, Hamilton, New Zealand, 2006.
68. Frank, E.; Xu, X.; Zealand, N. *Applying propositional Learning Algorithms to Multi-Instance Data*; Computer Science Working Papers; University of Waikato: Hamilton, New Zealand, 2003.
69. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
70. Sammut, C.; Webb, G.I. *Encyclopedia of Machine Learning*; Springer: Boston, MA, USA, 2010; p. 892.