PREDICTING STUDENT PERFORMANCE BASED ON LOGS IN MOODLE LMS

Mariela Mizota Tamada
Institute of Computing
Federal University of Amazonas (UFAM),
Manaus, Brazil
Institute Federal of Rondonia, (IFRO),
Porto Velho, Brazil
mariela.tamada@ifro.edu.br

Rafael Giusti
José Francisco de Magalhães Netto
Institute of Computing
Federal University of Amazonas (UFAM),
Manaus, Brazil
{ rgiusti, jnetto }@icomp.ufam.edu.br

Abstract*— Context: This innovative practice full paper presents a methodology to predict at-risk students in the context of a course assisted by an LMS (Learning Management System). LMSs generate large amounts of data about courses and students, which allows schools to make useful insights with the help of computational analytical tools. Most educational institutions claim that the most significant issue in virtual learning is high student dropout rates, and school performance is one of its main factors. Objective: Our study aims to use Machine Learning techniques based on logs from the Modular Object-Oriented Dynamic Learning Environment (Moodle). Those data are used to analyze student behavior and create a model that helps detect students at risk. Method: This paper used institutional data and trace data generated by LMS of a Computing education technical courses, blended and distance learning, at high school. We compared 7 algorithms with models trained at 6%, 20%, 40%, and 60% of the course duration, with the intent of exploring the compromise between early and late detection of at-risk students. Our model has 69% positive classe (failed) and 31% negative class (passed), and the false positives cost is important. Results: The results show 7 created models of predicting. The findings for Random Forest performed the best when predicting a student's performance. Conclusion: Our study provides a student at-risk prediction model using ML techniques on logs in Moodle LMS and may guide future studies and tool development to reduce these high dropout rates.

Keywords- performance prediction; Machine Learning; Learning Management Systems; logs in Moodle.

I. INTRODUCTION

Distance education has improved with the gradual advancement of information technology in the past decade, and to avoid school dropout, there is research on models for predicting academic performance. If the institution detects at-risk students in the early stages of the course, there will be more time to make an intervention that may improve their performance and help them complete the course successfully.

This work presents Machine Learning (ML) techniques for analysis of registration data and logs files of a Learning Management System (LMS), which is a virtual classroom environment. LMSs enables communication between teacher and students, and provides a space for students to access learning materials and course activities.

All interactions in the LMS generate a log, which stores information in a database. With this, the amount of data collected is rapidly increasing in volume and complexity, allowing statistical analysis, data mining, and building predictive models of school performance. Some surveys create predictive models with the log files generated in the course. Still, these models fail to generalize to an early prediction because the actual log information is different from that used to train the models. In this work, we use Machine Learning to predict academic performance in the LMS, analyzing the socio-economic profile and the LMS log files generated up to the forecast time. In other words, a model that fits the characteristics and the progress of the course in the passage of the first subjects.

Concurrent enrollment, more commonly known as dual enrollment, refers to programs where students are enrolled in two schools simultaneously in a high school environment, one in regular high school and another one in technical courses, offered in blended learning (b-learning) or distance learning (e-learning), which duration is 2 years. The educational institution analyzed has more than 600 campuses in Brazil, at different levels of education, ranging from integrated high school to graduate school, and courses take place in the traditional classroom, blended and distance learning formats. Technical education concomitant with high school is present in several of them.

In this work, the evaluated courses had, on average, 69% of failure or dropout, a high rate. Students who are unsuccessful in their studies lose time and effort in their failed searches, and they and their families can suffer financially and emotionally. Institutions also lose the scarce resources they have invested in.

The application of ML and statistics of information generated from an educational setting [1] and the knowledge discovered may help improve teaching/learning processes. This study analyzed ML techniques in LMS with the objective of early identification of students with high probability of evasion. Early identification of at-risk students may provide teachers, tutors, and managers with strategic information that helps in making decisions for appropriate pedagogical interventions.

These predictive models are essential to achieve equity in distance learning, a modality that has been growing in the last decade and has increased with the Covid-19 pandemic. Furthermore, we focus on courses concurrent with high school, that is, a level of education still little explored in research.

Based on the stated objective, we propose the following research questions.

RQ1. *Is it possible to predict students at risk by analyzing their LMS logs when 20%, 40%, and 60% of the course has been completed?*

RQ2*. Which are the variables derived from the LMS records that most influence student performance?*

The remainder of this paper is organized as follows: Section II presents the theoretical background, Section III presents related works, Section IV about tools, Section V details the research methodology, Section VI discusses the results, and offers an overview. Finally, section VII concludes the paper and presents further research and challenges of student performance prediction models using ML techniques.

II. Theoretical Background

This section presents two essential terms in this paper: the learning platforms used in distance learning and the ML concepts, which develop methods to discover patterns that lead to knowledge.

A. Learning Platforms

Among the platforms or LMS are Moodle[1], which is most commonly used to adapt traditional face-to-face courses to blended learning or entirely online courses, and MOOC[2] can simultaneously reach thousands of students in several countries, with no theoretical limit on attendance. However, the completion rate for most courses is below 13% [2]. The most cited reasons for dropout [3] are personal and workload reasons such as not enough time, increased workload at work, and the academic workload being evaluated as too high in some cases.

One feature of these courses is the large amount of data that can be collected from them. Besides the history and performance of the student, each action performed (reading files, participating in forums, sending messages, visiting recommended links, streaming clicks, time of activities) leaves a digital footprint, which allows for log analysis to follow their learning [4] or using mining tools to discover Association Rules used to identify dropout situations [5]. There are research fields dedicated to analyzing these digital education trails and Machine Learning is one of them.

B. Machine Learning

The theory of Machine Learning (ML) is a field that lies at the intersection of computer science, statistics, and mathematics, that has been subsequently adopted by researchers to predict student retention within virtual class environments [6]. With Big Data, ML research increases to learn from huge amounts of data. The prediction process consists of preparing the data, training the model with algorithms and variables until reaching the best performance and then using the model to predict with new data [7]. In the case of supervised learning, the data is labeled, which in our context means the data collected from the LMS is annotated with the student's performance. This is known as training data, which is the input information for the ML algorithm. Thus, the technique (algorithm) learns to identify from the experience of previous cases, with a certain rate of correctness or accuracy or other measures, the student's characteristics and determine (classify) whether or not they will drop out.

In this work, among the various measures to evaluate the tests, we used Precision, Recall, and F-measure. In statistical analysis of binary classification, the F1-score or F-measure is a measure of a test's accuracy. It is calculated from the Precision and Recall of the test.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Precision indicates how well a binary model can perform with respect to the positive class. Precision is especially important when the cost of False Positive is high.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Recall is the ratio of correctly predicted positive observations to the total of positive observations in the data. Recall is especially important when there is a high cost associated with False Negative.

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

Finally, F-measure gives a balance between Precision and Recall. F-measure is usually a good metric to assess the quality of a model when there is no particular reason to favor Precision or Recall and when there is an uneven class distribution in the data.

III. Related Work

In [8], the research analyzed 17 blended courses with 4,989 students on Moodle LMS, using logistic models (pass/fail) and standard regression (final grade). They predict student performance in the first 10 weeks.

In [9], they created Random Forest models to predict student success based on input predictors (lectures, quizzes, labs, and videos) extracted from Moodle records, with 96.3% accuracy. They study the dependency of predictors on the target value, finding that scores in laboratories and quizzes have the most significant influence on the final grade.

Some surveys create dropout prediction models based on a weekly analysis [10, 11, 12, 13, 14, 15, 16]. The authors in [11] explore the Deep Learning techniques by building a prediction model using data from the past weeks to predict if a student will drop out in the following week. In [15], one prediction model is created every week, and a model is also

[1] Modular Object-Oriented Dynamic Learning Environment, a free software created in 2001. moodle.org

[2] Massive Open Online Course, created in 2012 for free and open online courses. mooc.org

created with the full data. The weekly models are interesting in that they allow exploring changes in weight of each variable with the progression of the course. The authors concluded that, while some factors such as time spent watching videos become increasingly more important with time, the number of exercises solved is the most meaningful variable during the whole course.

Those papers have used supervised ML techniques for dropout prediction: LR (Logistic Regression) [10, 12], DL (Deep Learning) [11, 15], SVM (Support Vector Machine) [13], RF (Random Forest) [12] based on clickstream data, forum participation, and other resources. Furthermore, individual or combined tests with an excellent average accuracy of classifiers could present variations that follow the changes in a context in which learning is a dynamic environment, and many dropout prediction models are built weekly [17].

In [18], the authors studied the portability of predictive models of student performance among courses with the same degree and similar level of LMS usage. They create J48 decision tree models from 24 courses to classify students who pass or fail. The authors employ AUC[3] to measure the performance of the classifier. When trained and tested on the same courses, the AUC is high (above 0.80 in almost all cases), and when porting models between courses to the same degree, the AUC falls by 0.09 up to 0.28 points. These losses range from 0.22 to 0.25 when porting models between courses with a similar level of Moodle use.

In [19], they use machine learning to create models for the early prediction of students' performance in solving LMS assignments by just analyzing the LMS log files generated up to the moment of prediction. They predict students' performance at 10%, 25%, 33%, and 50% of the course length. MLP (multilayer perceptron) neural networks obtain 80.10% accuracy when 10% of the course has been delivered.

IV. TOOLS

Our model works through a PostgreSQL database server for processing the data with another interconnected tool, RapidMiner [20], a data science software platform for data preparation, machine learning, deep learning, text mining and predictive analytics, and has an exceptional GUI that attracts users. It also holds the advantage that it incorporates both R's and Weka's functionalities. It exhibited good operational characteristics and remarkable project activity characteristics [21].

V. METHODOLOGY

This paper analyzed the course with a forecast time in the 20%, 40%, and 60% of the course duration to detect students at risk of failing. Our goal is not to predict the exact grade but to detect students at risk of falling. To do this, we first provide an overview of the theoretical arguments used in learning analysis research and the typical predictors that have been used in recent studies. Then, we analyzed 8 classes of a technical course in the area of Informatics. Since raw data in log files is complex and fraught with noise, several methods are known for preprocessing the data source. Then, we create different classification models and analyze the defined metrics.

A. Dataset

The Federal Institute of Education, Science and Technology in Brazil tend to use the Moodle LMS. We analyzed data from a technical course concomitant to high school. This course is offered as blended teaching or distance education, and lasts two years. The curricular matrix is identical for technical courses after high school (adults), whether in traditional classrooms, blended, or distance education.

We selected 7 regular classes from the Informatics course for our analysis, starting in 2016 to 2018 and ending in 2020. Each class had up to 40 students enrolled. In addition, we also collected data from a special offer from 2017 with live online courses in 5 centers in the state, accompanied by tutors at each end. This class enrolled 500 students—all with the same curricular matrix and duration of 2 years.

The course consists of 3 modules, and each module has 3 to 5 stages in a row with two concurrent subjects in each step, totaling 25 subjects, which may have different hours. We analyze performance per student at each stage. Thus, 20% of the course duration has 5 subjects completed, 40% of the course has 9 subjects closed, and 60% has 15 subjects closed. It is not a weekly or monthly analysis but the advancement of disciplines at each stage. This criterion allows being independent of the duration and breaks of the course.

We used machine learning techniques on data from 761 students and 788,887 records extracted from the log to create performance prediction models. Because LMS logs are not designed for this type of statistical analysis, we must perform feature extraction on the logs. This produces the variables that ML models require to be trained. Such variables may include student engagement with the LMS, such as the number of assignments completed per day, week, month, etc., the frequency in which they access the teaching material, as well as their gradings as registered by the professor.

Students of completed courses have a final status, whether or not they have completed the course subjects. Together with the student report card, this information was in another system and was integrated via the CSV file.

The course has weekly synchronous classes (traditional classroom or remote), which at the time of the data collected was one weekly face-to-face meeting and from 2020, with the Covid-19 pandemic, became 2 remote weekly synchronous meetings.

B. Subset Selection

The data set was divided into 80% of instances for training and 20% for testing, using a stratified random sampling method

[3] The AUC (area under the roc curve) value lies between 0.5 to 1 where 0.5 denotes a bad classifier and 1 denotes an excellent classifier.

[22]. The data division is random and stratified to ensure that the proportions between classes are the same in each cycle.

As the raw data in log files are complex and full of noise, several criteria were used for data cleaning and we used several methods to pre-process the data source. Reasons for noise include missing data and incorrect registration.

Enrollment students who have no final status classification label comprise 8% of the dataset (830 instances); we chose to exclude them from the classification. The rest of the class comprises 761 students, of which 525 failed (69%) and 236 passed (31%).

We selected 14 variables (attributes) for the prediction model, of which 6 are institutional data (ID), and 7 are trace data (TD) [23]. Table I, the variable ID1 (age) as a high school, will only be between 14 and 19. The variable "Final result", which is the class attribute (intended for classification) in datasets, was dichotomized into "A" (passed) and "B" (failed), with the transformation of values "passed", "completed" and "necklace grade applications" into "A", and "withdraw", "dropout", "failed" into "B".

TABLE I. VARIABLES I OF THE MODEL.

Institutional data (ID)				
	name	type	description	values
ID1	age	continuous	variable calculated	14 to 19 years old
ID2	gender	categorical		
ID3	fam_income	continuous	family income in number of minimum wages	
ID4	am_fam_members	continuous	number of people in the family	
ID5	work_situation	categorical	family work situation	1-Unemployed, 2-Professional, 3-Employed, 4-Retired, 5-Entrepreneur, 6-Self-employed, 7-Cooperated.
ID6	fam_situation	categorical	income share	1 - Income Provider 2 - Dependent 3 - Make up the income
ID7	status	categorical	final status	"passed", "completed", "necklace grade applications", "withdraw", "dropout", "failed".

That failure refers to the non-completion of the course, which includes both dropouts and failures within the completion period, even though they are different concepts.

The attributes ID3, ID4, ID5, and ID6 are essential indicators for the public institution that offers these courses, as its target audience is students enrolled in regular high school from public and free schools. These variables had to be processed for missing data and incorrect values. For instance, in some cases ID3 (family income) was filled as units of minimum wage (*e.g.* "income equivalent to twice the minimum wage") or numbers were fully spelt as strings. Whenever possible, these values were parsed as numeric values, otherwise they were removed. Approximately 23% of the instances had missing value for this attribute, and they were kept as missing—models that cannot handle missing data ignore those instances.

Trace data (TD) was counted as a number of interactions with some LMS features. All variables are continuous. In Table II the variable TD1 (perPart) is the percentage of participation in interactions in the LMS, an attribute calculated with the rate of accesses in the subject about the accesses of the other students in the same class. For this, it disregards the interactions of actions 'view' (platform navigation only), 'login', 'logout', and 'mail error'. The variable TD2 (freq) is the frequency of courses offered in a blended and distance education format. In any case, there is 1 synchronous weekly meeting.

TABLE II. VARIABLES II OF THE MODEL.

Trace Data (TD)				
	name	type	description	values
TD1	perPart	continuous	accesses in the subject	percentage in relation of the other students in the same class
TD2	freq	continuous	list of presence of students	
TD3	gpaavgcourse	continuous	GPA (Grade Point Average)	average of the final grades of the subjects by the defined forecast time in the duration of the course.
TD4	avgsubjects	continuous	average grade of course activities	2 courses activities per discipline are required, in addition to regular assessment.
TD5	log_chat	continuous	participation in chat	sum of quantities of interactions
TD6	log_forum	continuous	forum participation	sum of quantities of interactions
TD7	log_quiz	continuous	participation in questionnaires	sum of quantities of interactions

Also, we analyzed access to URL (link to access external materials) and resources features, and the amount was a general average close to zero and, therefore, were disregarded.

Attribute selection is pretty important. The following metrics were used to select attributes and count the cost:

Correlation: Columns that too closely mirror the target column, or not at all;

ID-ness: Columns where nearly all values are different;

Stability: Columns where nearly all values are identical;

Missing: Columns with missing values.

In TD5 (log_chat) its missing value reaches 86.42%, the worst in this criterion and far from the 2nd worst which is 26.76%. In the case of ID5 (work_situation) it presents Correlation: 0.31%, ID_Ness 0.52%, Stability: 94.35%, Missing 26.11%; and ID6 (fam_situation) had correlation: 0.17%; ID_ness: 0.39%; Stability: 95.23%; Missing: 26.11%. They are not exactly like worst values in each criterion, but the resulting set informs that they are attributes that in this data context will not contribute to the analyzed prediction. The ID5, ID6, and TD5 had low-quality data, and were disregarded. Missing values for TD2 were replaced with zero.

C. *Data Analysis*

The RapidMiner tool uses a technical research grid that will thoroughly test all possible combinations of hyper parameters. It will provide input values and try all combinations by plotting a Cartesian plane (hence the grid name). Then, it will select the hyper parameters that obtained the minor error.

This tool shows the attributes, we select the feature for prediction, and it requests the class of most significant interest, which in our case is "B" = failing. Each attribute reports an information quality rate, based on the correlation, ID_ness,

stability, and missing data, to help our choice of relevant attributes and prepare the models with the selected algorithms.

We performed the algorithm and cross-validation 10 times for each dataset (blended and distance learning) and captured the total time spent in each run by comparing the seven classifiers. We performed all tests on Intel Core i7 7th Gen.-7.500 CPU, 2.90GHz, and 12Gb of RAM.

D. VI. Results

***RQ1**. Is it possible to predict students at risk by analyzing their LMS logs when 20%, 40%, and 60% of the course have been completed?*

Our intention is not to predict values (regression models) but to detect students at risk (classification models) to personalize, reinforce, and improve their learning. In order to see which students are at-risk or not, we model a binary classification problem (A = pass / B = fail). In the institution, the score for passing is 60 points. Both dropouts and students who failed to reach passing grades belong to class B.

We created different predictive models from the data set to answer RQ1. The grading limit defines a binary classifier. We considered a model with all classes and another only with the distance learning class.

In order to assess the quality of the prediction at different stages of the course, we built 3 binary classifiers, each using data collected at some point in the course completion (20%, 40%, and 60%). These data were used to train and test 7×3 classification algorithms. The variables were normalized to 0 and 1 since some classifiers such as MLP and SVM perform better with normalized attributes [24]. These models show the extent to which it is possible to predict student performance in the LMS at different forecast times.

We used the algorithms with default values (unless otherwise stated). The following algorithms were used: Random forest (RF), Decision tree (DT), Gradient Boosted Trees (GB), Logistic Regression (LR), Naive Bayes (NB), Deep Learning (DL), and Support Vector Machine (SVM). The models for predicting at later stages in the course have displayed better performance. The results demonstrate that the RF (Random Forest) obtained the best performance (Fig.1) on average.

The data sets include LMS interaction information and other data such as date of birth, gender, income, and others. The highest F-measure measures for the first measurement with 20% of the course progress were 84.32% and 91.78%. These values increase as the timing of the forecast increases.

We analyzed the data with the metrics F-measure, Precision, and Recall in the following cases:

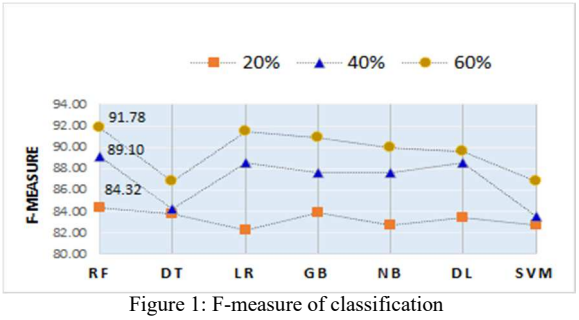

Figure 1: F-measure of classification

The RF (Random forest) had the best F-measure in the 3 analyzed instances (Fig.1) with 84.32% at 20%, 89.10% at 40%, and 91.78% +- 3.1% standard deviation at 60%, slightly better than the other algorithms. In the case of 20% the second, the best was the GB. It was the DL in the 40%, and in the 60%, it was the LR.

The optimal parameter for that RF is 20 trees and 7 as maximal depth and results in a 13.5% error rate.

Random forests are like the pulling together of decision tree algorithm efforts. They are taking the teamwork of many trees, thus improving the performance of a single random tree. Random decision forests correct for decision trees' habit of overfitting to their training set [25]. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance [7].

We have mentioned that the F-measure is the balance between Precision and Recall. When analyzing these, we found in Precision a situation with similar trends to the F-measure (Fig.2) but by a small margin.

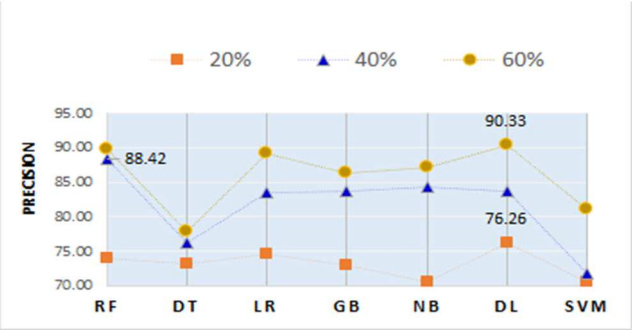

Figure 2: Precision of classification

Precision reflects the reliability of the model considering the positive class. Since our goal is to identify students at risk, we define the positive class as “B=failed”. High precision means that students deemed to be at risk of failing are indeed at risk or failing with high probability.

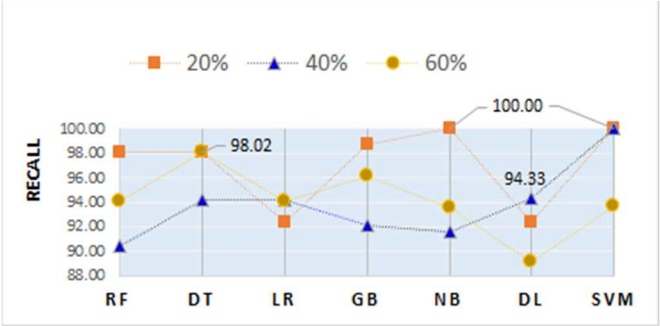

Figure 3: Recall of classification

However, Precision by itself is not the ideal measure. Recall (also known as Sensitivity) must also be taken into consideration. Recall is the ratio of correctly predicted positive observations to all observations in the actual class, and measures how reliable the model is in identifying most of the positive instances. Our models return high sensitivity (Fig.3). This essentially means that our model can identify students who are at risk of failing or dropping out.

We analyze the models in stages that correspond to 20%, 40%, and 60% of the advance in the course duration, and all the values of metrics result above 80%. For this reason, we decided to analyze the prediction with only the first completed subject, corresponding to the first 4 weeks of the course, and the only discipline of the first stage. All the following stages have 2 concurrent subjects.

RF obtained the best performance in the 3 models, and we decided to complement this research and check the model's behavior using only the first-course discipline, which corresponds to the first stage of the course and lasts 4 weeks. This corresponds to 6% of course completion. In this case, the RF fell to the worst result, and the LR had the best impact (Table III), represented in Fig.4.

TABLE III. F-MEASURE OF MODELS.

	Model			
	6%	**20%**	**40%**	**60%**
RF	77.58	**84.32**	**89.10**	**91.78**
DT	82.69	83.73	84.24	86.81
LR	**82.91**	82.24	88.55	91.48
GB	82.69	83.89	87.58	90.93
NB	82.59	82.69	87.68	90.01
DL	82.69	83.38	88.55	89.65
SVM	82.69	82.69	83.58	86.83

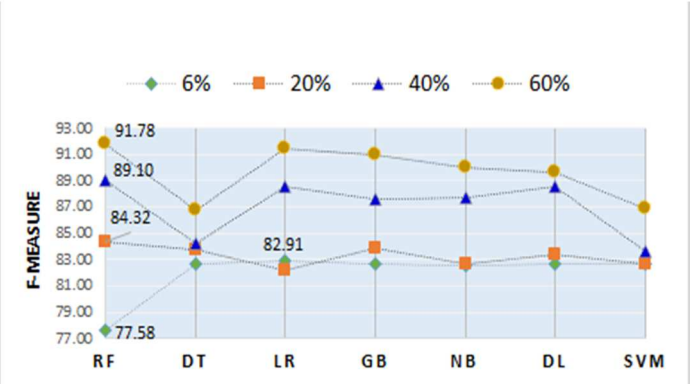

Figure 4: models with first stage

At the beginning of the course, there is still little data on interactions with the platform and explain the practically equal values in the performance of the algorithms, except for the poor performance of the RF. Only with a completed discipline of 25, the LR showed a slight advantage in this context. The course had 25 subjects in total. The models were tested at 6%, 20%, 40% and 60%.

The interpretation of the parameters of a LR model is obtained by comparing a probability of success with a likelihood of failure, using a function odds ratio - OR (odds ratio)[27], where p is the probability of success and q = 1- p is the probability of failure, so odds (failure) = q / p.

Next, we analyzed the 3 models corresponding to students enrolled in distance learning, with the same curriculum, considering the same metrics used in the hybrid teaching model: F-measure (Fig.5), Precision (Fig.6), and Recall (Fig.7).

RF continues to perform better at 20 and 60% of the course, but the DL emerges when it is at 40%.

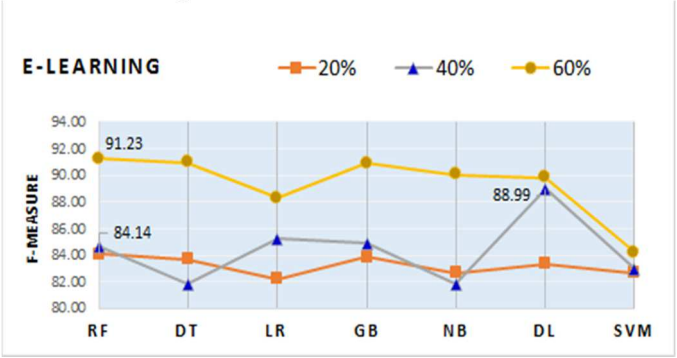

Figure 5: F-measure in distance learning course

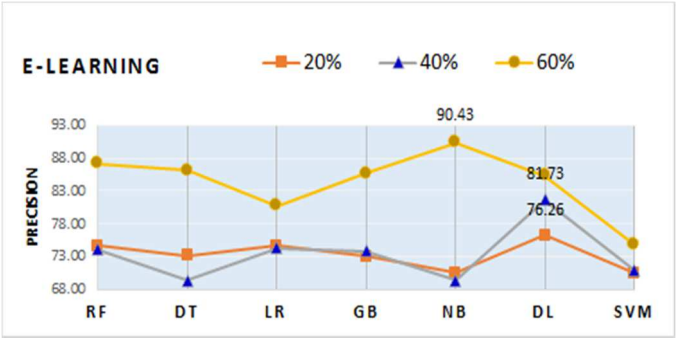

Figure 6: Precision in distance learning course

The NB algorithm goes from having the worst performance to being the best as the period progresses (Fig.6). It is a probabilistic classifier and completely disregards the correlation between variables (features).

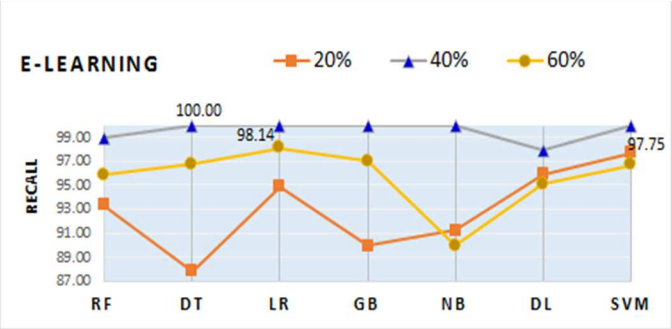

Figure 7: Recall in distance learning course

***RQ2**. Which are the variables derived from the LMS records that most influence student performance?*

In order to discover which variables contribute the most to evaluating the student performance, we analyzed the influence of each attribute to the classification model. Each attribute is assigned a weight according to its contribution to the decision, as shown in Fig. 8. We note that the most important attributes vary depending on the moment the analysis is performed.

20%

Attribute	Weight
gpaavgcourse	0.166
log_quiz	0.160
avgsubjects	0.160
freq	0.075
age	0.036
fam_income	0.034
gender	0.029
log_forum	0.021
am_fam_members	0.020
perpart	0.017

40%

Attribute	Weight
log_quiz	0.243
gpaavgcourse	0.179
freq	0.110
avgsubjects	0.083
fam_income	0.054
age	0.038
gender	0.029
am_fam_members	0.027
perpart	0.023
log_forum	0.017

Figure 8a: Weight of attributes in RF of the 20%, 40% duration.

60%

Attribute	Weight
log_quiz	0.255
gpaavgcourse	0.174
freq	0.114
fam_income	0.076
age	0.063
log_forum	0.050
avgsubjects	0.047
gender	0.040
am_fam_members	0.029
perpart	0.020

Figure 8b: Weight of attributes in RF of the 60% duration.

The average of the grades awarded in the first 5 subjects was the most relevant attribute for early detection (shown in Fig.8a-left for 20% course completion). Soon afterward comes the students' interactions with the questionnaires (log_quiz). Although Moodle has more than 10 types of resources (forum, questionnaires, task, wiki, URL, class, book, etc.) to diversify activities, teachers prefer to use the questionnaire as an evaluative activity, which we believe contributes to the importance of this variable. At 40% and 60% (Fig.8a, 8b) of the forecast time of the course, these 2 known characteristics were reversed and the difference between them is more pronounced. Questionnaire interactions are the most important subject for predicting student success. In part, this may be due to the fact that a significant motive for student failure is the lack of engagement with the activities.

The analysis of the variable, as seen in Figures 8a and 8b, also shows that trace data (TD) has a greater weight than institutional data (ID).

C. *Overview*

In this paper, we present strategies to identify at-risk students on different stages of course completion, namely at 20%, 40%, and 60% of 2-year Computer Science technical courses. Our data was collected from 8 classes of students: 7 b-learning and 1 e-learning. We also consider a very early prediction with data collected at 6% of course completion—at that point, students completed only one subject. So, 4 models with b-learning and e-learning, and 3 models with only e-learning.

We evaluated the performance of 7 Machine Learning (ML) algorithms. The RF (Random forest) showed the best F-measure in the 3 analyzed stages (Fig.1). We also analyzed the impact of each variable on the prediction and student interactions with activities that involve questionnaires on the LMS platform have shown to be the most relevant characteristic for predicting student success. Furthermore, the average of the grades was the most pertinent variable at the beginning of the course. This reinforces that students with a long history of little interaction with the system or who perform very poorly, in terms of grade, at the beginning of the course, should be considered potential dropouts.

Our findings suggest that tree-based models (DT and RF) are decent predictos for at-risk students with LMS platform data at later stages of the courses (40% and 60%). This is useful because tree-based models are relatively easy to interpret, and may allow not only to identify at-risk students with high confidence, but also to understand why they might be dropping out. For early prediction (20% or earlier), more robust methods such as Deep Learning neural networks are recommended.

D. Limitations

This study must be interpreted within the following limitations: (a) results may be subject to the automated engines of the RapidMiner tool.; (b) only studies written in English were selected as references; (c) this work was limited to one campus; and (d) there may be some bias involved in the features initially selected to create the data set from the logs and decisions to handle missing data.

VII. Conclusions and Future Work

The contribution of this work provides a student at risk performance prediction model using ML techniques on logs in Moodle LMS at high school courses, a less explored level of education. It may guide future studies and tool development to reduce the high dropout rates.

Thus, we created models of ML that can also contribute to other approaches and combine techniques and solutions.

It is intended to advance this research and be applied on campus with data for validation and future adjustments. This educational institution network with more than 600 campuses in the country that use the same LMS and could apply the model's effectiveness in students from the same degree and curriculum on other campuses.

Another survey with demographic analysis by campuses to detect differences in performance by geographic region would be complete.

We used the final status to create the model, but we do not have this label in ongoing classes. Therefore, an unsupervised approach would be significant, grouped by the behavioral characteristics in the LMS.

We analyzed student performance with a focus on the student at risk of dropping out. Also would be relevant clustering in 3 groups of students: at risk, regular and outstanding. Students who excel can also experience an intervention with challenges and contributions.

In the end, the model allows the adoption of measures that contribute to the personalized monitoring of students and the detection of students at school risk since it can serve as an instrument for making school intervention decisions.

VIII. Acknowledgments

We thank Coordination for the Improvement of Higher Education Personnel – CAPES Foundation. We also thank Federal Institute of Education, Science and Technology of Rondonia State –IFRO- for providing the database and administrative support. This research was partially supported by Amazonas State Research Support Foundation – FAPEAM and carried out within the scope of the Samsung-UFAM Project for Education and Research (SUPER), according to Article 48 of Decree nº 6.008/2006(SUFRAMA), was funded by Samsung Electronics of Amazonia Ltda., under the terms of Federal Law nº 8.387/1991, through agreement 001/2020, signed with Federal University of Amazonas and FAEPI, Brazil.

References

[1] S. Oeda and G. Hashimoto. "Log-Data Clustering Analysis for Dropout Prediction in Beginner Programming Classes". Procedia Computer Science, Vol. 112, 2017, pp. 614-621.

[2] D.F.O. Onah and J. Sinclair, R. Boyatt. "Dropout rates of massive open online courses: behavioral patterns". In: Proceedings of the Sixth International Conference on Education and New Learning Technologies, 2014, pp. 5825–5834.

[3] M. Morales, R. H. Rizzardini and C. Gütl, "Telescope, a MOOCs initiative in Latin America: Infrastructure, best practices, completion, and dropout analysis," 2014 IEEE Frontiers in Education Conference (FIE) Proceedings, 2014, pp. 1-7, doi: 10.1109/FIE.2014.7044103.

[4] S. Walldén and E. Makinen. "Educational Data Mining and Problem-Based Learning". Informatics in Education, 2014, Vol. 13, No. 1, pp. 141–156.

[5] F.A. Neto and A. Castro. "Elicited and mined rules for dropout prevention in online courses". 2015 IEEE Frontiers in Education Conference. Vol.1, pp. 1-7.

[6] M. Kloft, F. Stiehler, Z. Zheng, N. Pinkwart, K.D. Mattingly, M.C. Rice and Z.L. Berge. "Predicting MOOC dropout over weeks using machine learning methods". Knowl. Manag. ELearn.4, 60–65, 2014.

[7] S.B. Kotsiantis. "Supervised Machine Learning: A Review of Classification Techniques". IJCSIT, Vol. 7 (3), 2016, pp.1174-1179.

[8] R. Conijn, C. Snijders, A. Kleingeld and U. Matzat (2017). "Predicting student performance from LMS data: A comparison of 17 blended courses using Moodle LMS". IEEE Transactions on Learning Technologies, 10(1), 17–29.

[9] D. Ljubobratovic and M. Mateti. "Using LMS activity logs to predict student failure with random forest algorithms". 2019, The Future of Information Sciences, 113.

[10] C. Burgos, M.L. Campanario, D.D.L. Peña, J.A. Lara, D. Lizcano and M.A. Martínez. "Data mining for modeling students performance: A tutoring action plan to prevent academic dropout". Computers & Electrical Engineering, v.66, 2018, pp. 542-556.

[11] W. Xing and D. Du. "Dropout Prediction in MOOCs: Using Deep Learning for Personalized Intervention". Journal of Educational Computing Research. I-24, 2018, doi: 10.1177/0735633118757015

[12] X. Lu and S. Wang, J. Huang, W. Chen, Z. Yan. "What decides the dropout in MOOCS?". DASFAA 2017 Workshops, LNCS 10179, pp. 316–327, 2017

[13] M. Fei and D. Yeung. "Temporal Models for Predicting Student Dropout in Massive Open Online Courses". 2015 IEEE 15th International Conference on Data Mining Workshops.

[14] J.K.T. Tang, H. Xie and TL. Wong. "A Big Data Framework for Early Identification of Dropout Students in MOOC". In: Lam J., Ng K., Cheung S., Wong T., Li K., Wang F. (eds) Technology in Education. Technology-Mediated Proactive Learning. Communications in Computer and Information Science, vol 559, 2015, Springer, Berlin.

[15] C. Isidro, R.M. Carro and Ortigosa. "A. Dropout detection in MOOCs: An exploratory analysis". SIIE 2018 - 2018 International Symposium on Computers in Education.

[16] M. Teruel, L. A. Alemany. "Co-embeddings for Student Modeling in Virtual Learning Environments". 2018, 26th Conference on User Modeling, Adaptation and Personalization.

[17] M.M. Tamada, J.F.M. Netto. “Predicting and Reducing Dropout in Virtual Learning using Machine Learning Techniques: A Systematic Review”. 2019 IEEE Frontiers in Education Conference (FIE). doi: 10.1109/FIE43999.2019.9028545

[18] J. López-Zambrano, J. A. Lara and C. Romero. "Towards portability of models for predicting students’ final performance in university courses starting from Moodle logs". 2020, Applied Sciences, 10(1), 354.

[19] M. Riestra-González, M. d. P. Paule-Ruíz, F. Ortin. "Massive LMS log data analysis for the early prediction of course-agnostic student performance", Computers & Education 163. 2021, 104108, Elsevier, 2020.

[20] RapidMiner. Software of Data Mining: University of Dortmund: Rhineland-Westphalia, Germany, 2020; Available online: https://rapidminer.com/us (accessed on 25 Jan. 2021).

[21] P. Barlas, I. Lanning and C. Heavey. "A survey of open source data science tools". International Journal of Intelligent Computing and Cybernetics. 2015. ISSN: 1756-378X

[22] Z. Reitermanová. "Data Splitting". WDS'10 Proceedings of Contributed Papers, Part I, 31–36, 2010. ISBN 978-80-7378-139-2 © MATFYZPRESS.

[23] P. D. N. Silveira, D. Cury, C. Menezes and O. L. dos Santos, "Analysis of classifiers in a predictive model of academic success or failure for institutional and trace data," 2019 IEEE Frontiers in Education Conference (FIE), 2019, pp. 1-8, doi: 10.1109/FIE43999.2019.9028618.

[24] B. K. Singh, K. Verma, A. S. Thoke, "Investigations on impact of feature normalization techniques on classifier’s performance in breast tumor classification", International Journal of Computer Application 116 (19).

[25] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2008). "The Elements of Statistical Learning" (2nd ed.). Springer. ISBN 0-387-95284-5.