This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2911608, IEEE Transactions on Learning Technologies

From Lab to Production: Lessons Learnt and Real-Life Challenges of an Early Student-Dropout Prevention System

Alvaro Ortigosa, Rosa M. Carro, *UAM*, Javier Bravo-Agapito, David Lizcano, *UDIMA,*
Juan J. Alcolea, Oscar Blanco, *DIMETRICAL*

***Abstract*— This paper presents the work done to support student dropout risk prevention in a real online e-learning environment: A Spanish distance university with thousands of undergraduate students. The main goal is to prevent students from abandoning the university by means of retention actions focused on the most at-risk students, trying to maximize the effectiveness of institutional efforts in this direction. With this purpose, we generated predictive models based on the C5.0 algorithm using data from more than 11,000 students collected along five years. Then we developed SPA, an early warning system that uses these models to generate static early dropout-risk predictions and dynamic periodically updated ones. It also supports the recording of the resulting retention-oriented interventions for further analysis. SPA is in production since 2017 and is currently in its fourth semester of continuous use. It has calculated more than 117,000 risk scores to predict the dropout risk of more than 5,700 students. About 13,000 retention actions have been recorded. The white-box predictive models used in production provided reasonably good results, very close to those obtained in the laboratory. On the way from research to production, we faced several challenges that needed to be effectively addressed in order to be successful. In this paper, we share the challenges faced and the lessons learnt during this process. We hope this helps those who wish to cross the road from predictive modelling with potential value to the exploitation of complete dropout prevention systems that provide sustained value in real production scenarios.**

***Index Terms*—Educational data mining, e-learning, prediction methods, student dropout, warning systems**

I. Introduction

WEB-BASED educational models have consolidated during the last years. Many institutions use Learning Management Systems (LMS) as a complement to face-to-face instruction [1],[2] and in many cases courses are conducted entirely online [3],[4]. For example, the National Distance Education University in Spain provides more than 600 online courses to more than 200,000 students [5] and the Open University in the United Kingdom serves more than 173,000 students through online courses [6]. In the context of distance learning, high dropout rates are a well-known problem. In Spain, for example, distance learning has a dropout rate around 60% whilst face-to-face education reaches 24%, according to the Spanish Ministry of Education [7].

Being able to detect empirically and as early as possible those students who are at potential risk of dropping out is essential for maximizing the effectiveness of institutional retention efforts. It allows institutions to intervene in a timely manner by taking actions aimed at preventing dropout, as well as to focus their available (and generally scarce) resources on the neediest subpopulation.

Educational Data mining (EDM) techniques have proved to be useful in this context [8],[9]. Some works focus on predicting dropout at the course level [2],[10],[11], others at the degree level [12],[13], but few at the institutional level. In spanish distance universities, about 10% of students drop out of one degree to enroll in another [7]. These cases are not considered "dropouts" at the institutional level. Our research focuses on institutional level dropout. It has taken place at UDIMA, a Spanish university in which courses are conducted entirely online. It offers undergraduate and graduate courses of different areas such as Law, Criminology, Computer Science, Business Administration, Economy, History, Psychology and Education. The main goal is to predict (and intervene in order to prevent) situations in which a student that has not completed his degree does not enroll in any course from either the same or a different degree at the university in the following academic year. This situation is what we refer to as "dropout" in this work. We have created SPA, a system to predict and prevent dropout that is novel in several ways, since it combines the following characteristics:

--It supports multiple and updated predictions: the system delivers a very early initial prediction for each student, right after enrollment, as well as predictions updated dynamically to incorporate all the new data available, periodically throughout the whole academic year.

--It focuses on institutional dropout: the system evaluates

This work was supported in part by the Regional Government of Madrid, project e-Madrid-CM: Investigación y Desarrollo de Tecnologías Educativas en la Comunidad de Madrid (P2018/TCS-4307).

A. Ortigosa and R.M. Carro are with the Department of Computer Science, Universidad Autonoma de Madrid, Madrid 28049, Spain (e-mails: alvaro.ortigosa@uam.es, rosa.carro@uam.es)

J. Bravo-Agapito and D. Lizcano are with the Department of Computer Science, Madrid Open University, Collado Villalba 28400, Spain (e-mails: javier.bravo@udima.es, david.lizcano@udima.es)

J.J. Alcolea and O.Blanco are with DIMETRICAL, The Analytics Lab, Alcorcón 28923, Spain (e-mails: jjalcolea@dimetrical.es, ojblanco@dimetrical.es)

1939-1382 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2911608, IEEE Transactions on Learning Technologies

the risk of leaving the institution, not a certain course or degree.

--It uses multiple data sources: we do not use a unique source of information about the students, but combine data from different institutional sources, including Moodle LMS and administrative databases.

--A big data set is used for training the models: real data from the interactions of about 11,000 students along a 5-year timespan have been used to train the predictive models.

--It goes beyond predictions: the system not only provides predictions, but also supports the recording and inspection of all the retention actions taken, in order to evaluate their effectiveness later.

--It is a live system, deployed in a real environment: The system is in production since 2017 on a large scale in a real distance university, supporting dropout prevention in all the undergraduate courses offered by this university; at the time of writing this paper, it has been used during 3 semesters for about 5,700 students.

In this article, we not only describe the steps taken, the predictive models generated, the system developed and the results obtained in terms of predictions, but we also share the experiences lived, the challenges faced and the lessons learnt in this journey from the lab to production, i.e., from the potential value of the predictive models generated and tested in the lab during the research and prototyping stages, to their real and sustained value for student retention in a real production scenario.

The paper is structured as follows: section II presents the state of the art; section III describes SPA, the dropout prevention system; section IV shows the details of the predictive models generated; section V presents the use of SPA and the results obtained; section VI describes the challenges and lessons learnt during the whole process; and, finally, section VII comprises the conclusions and future work.

II. State Of The Art

The development of Early Warning Systems (EWSs), able to detect and warn about the risk that a student drops out, has long been a challenge, even in the context of face-to-face education. For example, in [14] the authors provided a guide to develop EWSs for high school, based on indicators such as student attendance and performance. In [15], the student disengagement was attributed to both individual factors (such as attendance, behavior and course performance) and institutional factors (such as school resources, demographic composition or personal relationships among instructors and students). Differently, [16] found out that academic efficacy and academic apathy were the best predictors of students at risk of receiving poor grades. Going beyond, in [30], once the data were analyzed, three types of actions were proposed: direct action on the student, action by interest groups (mixing students prone to abandonment with bright students who can help them) and action on legal parents.

In the e-learning context, EDM techniques have been widely used [8],[9] to support the prediction of different issues (such as student failure or dropout) on which EWSs can be built. As it has been said before, they are very useful to predict different issues such as student failure or dropout. Regarding the information commonly used to build predictive models, in many cases academic grades and attendance have been considered [17]. Information about the student background, his interactions within the LMS and the results obtained in continuous assessment is used in [11]. In [18], data about the students' age, gender, distance from home, pre-enrolment and first term performance are used. In [13], both academic and social data are combined with predictive purposes. Most of these works make use of information generated while the students are taking the courses, which may not be available for earlier predictions.

In other cases, the models do not include this type of information, but basic administrative data along with additional ones to improve the quality of prediction (e.g., periodic national exams for primary school students, or household surveys and census data for older ones) [19]. In [1], the authors make emphasis on the need of considering other sources of data beyond the LMS records to improve early predictions, such as personality features [20], learning styles or motivation [21]. They analysed 17 blended courses and the inconsistencies found on the results obtained made it difficult to draw general conclusions about the online behavior of potential students at risk [1]. In our work, we combine information from different sources, including all the data available in administrative databases from the very beginning along with all the interactions registered within the LMS.

Dropout prevention has been attempted at different educational levels. For example, the Wisconsin Dropout Early Warning System (DEWS) assesses the individual risk of failure to graduate on time for students in public K-12 schools [17]. In high school contexts, several experiments have taken place, such as the ones described in [22] to predict dropout at different steps of a course; the algorithms used in this work are able to predict dropout within the first 4-6 weeks of student enrollment. In higher educational contexts, dropout has also been predicted, mainly at course level [2],[10],[11] and sometimes at degree level [12],[13]. In the case of [12], models were built using information collected at three different moments throughout the first semester of the students' first university year.

Some of the works focused on preventing dropout aim at selecting the best model to early predict students at risk of failure or drop out. For example, in [11] methods based on decision trees (BART and Random Forest) performed better than the others. In [18], Random Forest and Classification and Regression Trees (CART) led to the best results. Bayesian networks have also been utilized: in [13] the K2 algorithm generated the model that best fit the data. In [23] the authors investigate whether semi-supervised algorithms (Self-Training, Tri-Training, Co-Training, De-Tri-Training, RAS-CO, and Rel-RASCO) could be useful to predict dropout in distance higher education. They compared the results got with those from C4.5 and Naive Bayes algorithms, and they found that Tri-Training algorithm performed better than the others.

In many studies focused on early prediction, the plan is to support retention actions and interventions as future work [22]. Studies assessing the efficiency of retention actions or

1939-1382 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2911608, IEEE Transactions on Learning Technologies

strategies, in the case that they are taken, remain scarce [24].

Most of the articles published in the context of early dropout prediction report works focused on specific courses [2],[10],[11] or, at most, certain degrees [12],[13]. However, up to our knowledge, the focus is seldom on preventing dropout at institutional level, i.e., focusing on retaining students in the institution, enrolled on courses of either the same degree or a different one.

In addition, in few cases the research results have been put into production in real live systems on a large scale. For example, the system presented in [17] is a massive system in production to provide early warnings. The main differences with our work is that it is used in a different context than higher education, predictions are made with a false positive rate that can reach 60%, and no updated predictions are generated throughout the course. In the case of [25], the context is also different from higher education and the data for scoring has to be provided by each school. Finally, there is a lack of articles reporting the difficulties and challenges that arise when moving from lab research to production, or giving useful advice for transferring research results into a real production environment. This is the main gap we intend to fill with this work.

III. The Dropout Prevention System

A. Problem, Goals and Definitions

The problem addressed can be summarized as follows: i) distance education suffers from high dropout rates (60% in Spain [7]); ii) dropout can be prevented through personalized retention actions aimed at specific students at risk; and iii) carrying out personalized retention actions requires the effort of professionals (counsellors/tutors), which are scarce in comparison with the number of students taking the courses.

In this context, the main objective of our work is to identify the most at-risk students so that the scarce advisory resources can preferably focus on them as early as possible, thus increasing the effectiveness of institutional retention efforts.

There are some secondary goals that complement the previous one: i) to understand the dropout risk factors in order to shed light on the possible causes, so that more effective and informed dropout prevention policies can be defined; ii) to keep track of the retention actions taken, so that their effectiveness can be analyzed later.

In order to understand the context of this work properly, it is important to define some concepts and terms:

Academic year: In Spain, undergraduate studies last four academic years, each of them composed of two consecutive periods (semesters). One academic year is named according to the two calendar years it embraces (for example, the current academic year in Spain is "2018-19").

Dropout: A case of dropout refers to the situation in which a student that, without having completed his degree, does not enrol at the university in the following academic year. Our targets, therefore, are students at risk of leaving the university (not only a specific subject or a particular degree).

New students: freshmen, i.e., students who are joining the university for the first time.

Recurrent students: not novices, i.e., students who have enrolled in courses in previous academic years.

B. The System and its Architecture

In order to reach the goals stated before, after a research stage to assess its viability, we developed SPA (Spanish acronym for Dropout Prevention System) in 2016, in the framework of a collaboration project between the University and a Spanish EdTech Startup. The key functionality of this EWS summarizes as follows:

--Delivering informed dropout risk predictions to users (tutors/counsellors) for every student as early as possible, at enrollment time as well as in predefined milestones throughout the academic year. It uses data available at the institution and at Moodle LMS, along with predictive models, to generate the predicted dropout risk value for each student.

--Registering all the retention actions taken on each student to prevent dropout.

Fig. 1 shows the system architecture, composed by four main modules: the extraction/transformation/load engine (ETL), the model generation framework, the scoring engine and the web application. More details on each module are given next.

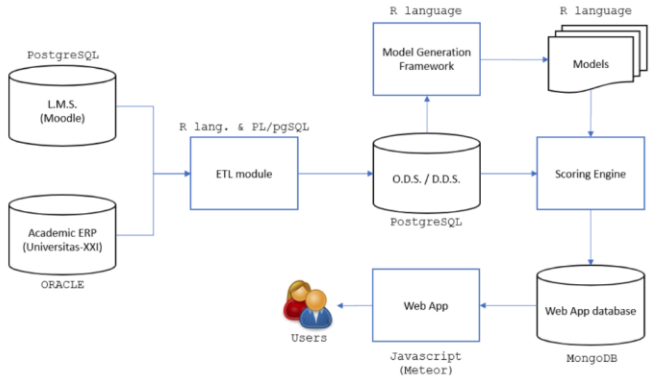


Fig. 1. Architecture of SPA.

1) The ETL Engine

It is responsible of: 1) extracting data from the sources and loading them into the Operational Data Store (ODS); the ODS is used to hold the temporary copies of the source tables and the intermediate results generated when calculating the features needed for subsequent model training/scoring processes; 2) transforming the fine-grained, detailed data loaded into the ODS into the complex, aggregated features used by the predictive models; and 3) loading the calculated features in the destination Decisional Data Store (DDS). The DDS is used to store the final values of the features resulting from the complex transformations carried out on the data from the ODS.

It is worth mentioning some aspects of the ETL process. The engine generates exact copies of the source system tables in the ODS using simple SQL select sentences to limit the workload generated in the source databases (which are part of active and mission-critical academic systems). The heavy transformation workload occurs within the ODS database, isolated from the source systems, through R and PL/pgSQL code that transforms the low-level data copied from the source systems into appropriate derived characteristics. These features are then

1939-1382 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2911608, IEEE Transactions on Learning Technologies

loaded into the DDS. The approach followed to make these copies varies depending on the amount of information to transfer. For example, LMS log data, which are larger in orders of magnitude than any other data source table, are loaded following an incremental approach. The whole ETL module is coded combining R language with PL/pgSQL stored procedures. This module is about 2.5 KLOC (thousands of lines of code) in size.

2) The Model Generation Framework

It provides the functionality needed, every academic year, for new model training, model evaluation, generation of model graphical representations, model parsing, scoring-code generation, etc.

Model parsing and code generation are necessary because the direct application of the generated models, in their native form, does not meet the model explainability requirement, necessary to achieve the secondary goals described above (they would yield the risk values for each student with no explanations about the reasons for each of them). Hence, those native R objects are parsed and translated into enriched base-R source code containing all the relevant information for model transparency, which is presented to the final user.

These enriched versions of the models allow the scoring engine to provide to the final user: 1) for each score generated for every student, information about which features were evaluated, and their exact values for each individual; and 2) the impact of each feature on the student's final dropout risk (i.e., whether it increases or attenuates risk) and its impact value (the intensity in which it increases/attenuates the risk). This module is about 0.5 KLOC in size.

3) The Scoring Engine

It generates, periodically (in each milestone throughout the academic year), a dropout risk value for each student, along with the corresponding explanations, according to the results obtained when feeding the proper enriched models generated using the previous module with the features generated by the ETL engine and stored in the DDS for each period. The resulting data are stored in the web app database and become ready for the end users to access it. The scoring engine is completely coded in R with SQL code embedded. The whole module is about 1.5 KLOC.

4) The Web Application

The Web application supports the interface between the system and the final users. Through this application, the users can access both aggregated and detailed information of their students' dropout risk along with the retention actions carried out for each of them. They can also register the retention actions taken by themselves. The web application is coded in JavaScript on Node.js using the "Meteor" development framework. The whole app is about 2.0 KLOC.

IV. The Predictive Models

As exposed above, the Model Generation Framework is used to build several models that are later embedded in the scoring engine. These models are the core of the system, and deserve a more detailed explanation.

A. Input Data

The data used to train the models is got and integrated from two different systems: i) UNIVERSITAS-XXI, the academic management system, a commercial ORACLE-based ERP for Higher Education [26] and ii) Moodle, the very well-known and widely used open source learning management system [27].

The former is a great source of static, general, administrative and academic information, together with some socio-demographic data. The latter provides detailed information on all the activity and the interactions of the students in their learning context, i.e., while taking the courses. When the project started, the institution had complete data for about seven full academic years on both systems. For each student, the data used to train the models falls into one of these categories:

--Personal information: age, gender.

--University access type: previous studies that allowed the student to enroll the university (high school, vocational training, elderly programs, etc.).

--Enrollment: semester of enrollment, number of credits and courses the student has enrolled for, type of credits/courses (core, compulsory or elective), course semester, number of credits and courses retaken by the student (taken more than once) and number of times he has taken each one.

--Economic/administrative data: type of fee payment (fragmented or unique), early/late enrollment and type of discounts applied.

--Academic results data (from previous academic years): percentage of degree completed, exam attendance ratio, exam success ratio, performance rate (number of credits passed from the ones enrolled) and average grade.

--LMS activity habits: percentage of activity by type of day (working/festive) and period of day (morning, afternoon, evening, night, etc.)

--LMS communications: number and average length of the messages sent to/received from peers and teachers.

--LMS activity levels: numbers of events recorded, posts written, discussions created, discussions accessed; tasks submitted, tests submitted and courses accessed.

--LMS academic results: grades obtained in tests and tasks, difference (in days) of each task submission date regarding the median of their peers', percentage of tasks completed.

All these data are processed to generate about 120 derived features describing each student in each period to be used later, in the predictive model generation and scoring processes. Some aspects of feature generation are worth mentioning:

1) Absolute/relative measures

When possible, each feature is calculated in both absolute and relative forms. That is, as an isolated datum describing an absolute aspect of a certain student in a certain period, and as comparative data describing his position with respect to his peers on that same aspect and period. For example, the number of posts that a student has written in forums in the last period is recorded. In addition, a value is calculated to represent how this student qualifies in terms of the number of posts written regarding those written by his peers, expressed in terms of a percentile.

1939-1382 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2911608, IEEE Transactions on Learning Technologies

2) Normalization

Some administrative information is coded in the source system as a range of values much wider than needed for our purposes. Therefore, the classifications are very disperse and scattered, with a high number of different values representing the same overall reality with non-relevant (for our goal) administrative variations. In these cases, a normalization process is performed to map this multiplicity of values to a much narrower and meaningful range for our purpose.

3) Aggregation

Some features are calculated at the course level, and must be aggregated and simplified into a single value per student/period to feed the models. In these cases, several summary values are generated to keep as much information as possible (mean, median, min, max, standard deviation).

4) Reflecting change

For each numerical base feature, two new ones are generated to reflect short-term/long-term changes: 1) short term: difference between the accumulated values of the current period and those from the previous one, and 2) long term: slope of the regression line for all the measures since the first period (to encode the trend sign and the change intensity).

The web application also includes some extra data extracted from the source systems. These student's data are not used to feed the models but to generate contextual information for the tutor/counsellor such as his name, contact information, self-description as written in his LMS user profile, etc.

B. Model Generation

Several strategies and approaches can be followed when facing the task of developing dropout prediction models. In order to ease the understanding of the approach taken, three key early decisions that strongly affect the final design are explained next:

1) New students and recurrent students are considered separately, as different populations with different problems.

We found two relevant facts in the research stage, when studying and analysing the reality to be modeled: i) dropout rates for new students are consistently much higher than for recurrent students (up to three times higher); ii) there is usable and relevant information about recurrent students that, by definition, does not exist for new ones.

While the event to predict is the same (dropout), based on those facts we decided to split the original problem of predicting undergraduate student dropout in two, in order to allow the generation of specialized models: i) predicting dropout of new undergraduate students and ii) predicting dropout of recurrent undergraduate students.

Both, early predictions and ongoing updated predictions are necessary to support retention.

The retention effort must start as soon as possible and must last as long as the full academic cycle. Therefore, the system must be able to generate both the earliest possible predictions and periodic updated predictions based on each student's changing behaviour and results obtained throughout the academic year. With this goal, the task is split into several separate subtasks, as follows:

Specialized models are developed to generate the earliest possible predictions. They use the scarce information available just after the student enrolls at the beginning of the academic cycle, generating an early risk estimation value even before the course begins. We call these models static, because the information they use is quite stable and does not include data on activity in the LMS, since no activity has occurred yet. With these models, we can calculate a first, very early risk prediction as soon as possible: the same day a student enrolls.

The entire academic cycle is divided into periods (usually, 15 days or 1 month each). At the end of each one, a new updated prediction is made for every student, using all the information generated in the last period along with all the information that was previously available. Therefore, N models are generated to predict dropout at N specific moments throughout the academic year. We call these models dynamic because, unlike the early static models, they use new and constantly updated information collected from the LMS.

2) Numerical dropout risk values are generated, instead of absolute YES/NO dropout predictions.

The underlying problem deals with the optimum use of limited resources to provide support to at-risk students. In these circumstances, having numerical values of dropout risk allows for prioritizing support: students may be served in descending order of risk score until support resources are depleted.

These decisions give rise to a scenario in which the task to be done is divided into several smaller and specialized subtasks, based on these criteria: type of student, that is, new/recurring students, and type of prediction, that is, static (unique, early)/dynamic (recurring, time milestone-based). Therefore, a specialized predictive model is required for each combination of criteria. Dividing the academic year into monthly periods from September to June (ten periods), the system requires the generation of 22 models: 2 for static early predictions (one per each type of student) and 20 for dynamic predictions (one per period per each type of student).

The explainability requirement, i.e., the need of not only giving a risk prediction but also explaining why, constrained us to use only white box models. Nevertheless, in the research stage we used Random Forest (RF), a well-known black-box technique that has proven to perform well in dropout prediction problems [28], to set an approximate upper bound in terms of model performance, since black-box algorithms usually perform better than the white-box ones required in production.

From all the data available, we reserved 30% for validation. For the first generation of 22 RF models, we obtained an average sensitivity of 65.5% on validation data, at a fixed false positive rate of 20%. This value was judged as the maximum acceptable false-positive rate for practical reasons and is considered the reference value for comparisons (see [29] for a discussion on the importance of proper model performance metrics). ROC curves were generated for each of the 22 models. Fig. 2 shows the ROC curve corresponding to the earliest (at enrollment time) RF model performance for recurrent students on validation data. Table I summarizes the performance of this model on the validation data used, setting a false positive rate of 20%, for both types of students in all periods.

1939-1382 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2911608, IEEE Transactions on Learning Technologies

In general, two trends appear when comparing model performances:

--Model performance increases as the academic year progresses. This matches intuition, since more information becomes available for the models as time passes and students interact with the LMS, and behaviours leading to future dropouts become increasingly evident.

--Models for recurrent students outperform models for new students. Again, this coincides with intuition, since models for recurrent students can take advantage of relevant information about student performance and interactions in previous academic years, not available for new students.

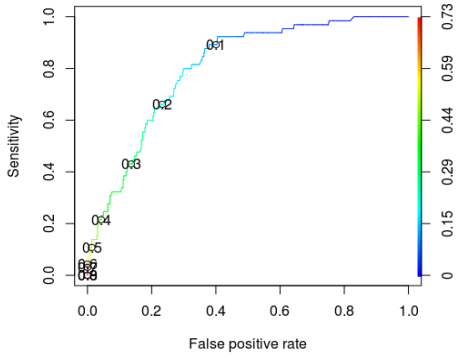


Fig 2. ROC curve: earliest (at enrollment time) RF model performance for recurrent students (on validation data).

TABLE I
SENSITIVITY OF THE RANDOM FOREST MODELS AT A FIXED 20% FALSE POSITIVE RATE, BY PERIOD AND TYPE OF STUDENT

Sensit.	New	Recurrent
Enrollment (Period 0)	38%	61%
Period 1	42%	62%
Period 2	46%	63%
Period 3	51%	64%
Period 4	55%	68%
Period 5	63%	80%
Period 6	65%	80%
Period 7	68%	80%
Period 8	71%	81%
Period 9	74%	82%
Period 10	76%	71%

Considering all the above, we chose the C5.0 algorithm to generate the final production models for the system, because: it is white box and easy to interpret; it is able to deal with problems of unbalanced binary classification (such as dropout); it is capable of generating probabilities in addition to absolute classifications; it is able to deal with quantitative and qualitative features; it does not require too much computing power during model generation and scoring; and its family of algorithms has shown to perform well in similar scenarios (see [18] or [23]). During model generation, we had to face several challenges. The most complex and/or time-consuming were:

1) Processing huge volumes of data.

Even though more historical data were available in the institution, we decided to limit the period used to train the models to the last 5 years. The reason is that, in general, the older the data, the more obsolete the realities they describe and, therefore, the less useful they are to predict the future. The 5-year limit was set to find a balance between "valid for prediction" and "sufficient" data. With this limit, we obtain an approximate number of 11,000 training samples, of which 30% correspond to new students, and the remaining 70% to recurrent students. The volume of data related to these 11,000 cases is huge: hundreds of millions of records spread over dozens of database tables, which is equivalent to approximately 50 GB of information. Processing this amount of data to generate the model characteristics for the training task requires a lot of computing and storage power. Some of the more complex ETL (Extraction - Transformation - Load) processes during model training take about 6-8 hours to run and complete in a 2-proccessor (Xeon E5645), 16gb RAM, Linux server.

2) Model tuning.

Most of the machine-learning algorithms can be adjusted though a set of parameters to generate models more tailored to certain specific conditions of the training data, and, as a result, performing better. C5.0 is no different. Many parameters had to be adjusted using ten-fold cross validation ten times to compare model performance. The most important/complex ones in our case were related to: a) providing an adequate cost-matrix to deal with the unbalanced nature of the dropout problem (one class is much more prevalent than the other: 29/71 for new students and 12/88 for recurrent students) and striking a good balance between sensitivity/specificity, and b) avoiding over-fitting, by limiting the depth of the trees generated by setting a minimum number of training cases in leaves of the trees to maintain the generalization properties of the model.

Even though the model training processes were fed with the full set of calculated characteristics, each of the 22 models selected only a particular subset of them (not necessarily coincident) to generate the predictive logic. When analyzing the logic of the generated models, in general terms, the following conclusions are drawn: In static models for early prediction, the features that dominate the rankings for new students are, in order of importance: age, university access type, number and type of credits the student has enrolled for and discounts applied. The relevant features for recurrent students are: performance rate, number, type and distribution of enrolled credits, percentage of degree completion and number of credits in re-taken courses. In periodic dynamic models, the features that tend to dominate the rankings, regardless of whether the model is for new or recurrent students, are: 1) LMS Activity-related features (the total, accumulated amount of activity since the start of the academic year; the general activity registered in the last period (month); and the amount of a certain specific activity (ie. forum posting) in the last period (month); 2) comparative features (how each student compares to his peers regarding the number of tests/tasks submitted and the grades obtained); and 3) student's workload and course distribution features (number and type of courses/credits enrolled, distribution of courses/credits along the academic year).

We have found some notorious differences between the features and, specially, their importance in the models generated for new students versus those for recurrent students. For new students, age and university access type are relevant

1939-1382 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2911608, IEEE Transactions on Learning Technologies

features. For recurrent students, although these features are also available, they are systematically ignored by the models. For recurrent students, many of the specific features that are not available for new students are selected by the models and tend to rank high regarding variable importance. They are all related to performance in previous years: exam presentation rate, percentage of degree completion, number of sabbatical years taken and number of credits retaken.

V. Use and Results

A. System Operations

1) Administrators

From the system administrators' point of view, one of the most important tasks is the periodic generation of updated risk scores. The system provides an interface that, considering the current date, lets the administrator launch the proper scoring process. This happens daily during the initial enrollment period, to include new enrollments and to update, if necessary, scores already calculated, since new enrollments occur constantly, and existing ones can also be modified in this period.

Once the enrollment period is finished, the scoring is run once per period. Usually, the academic year is divided in ten 1-month periods, from September to June. Therefore, the scoring processes are run monthly, generating updated risk information the first day of every month. The scoring process takes between two and three hours to run and extracts about ten million records from the source systems. At the end of the academic year, about 20 GB of data have been generated.

It is important to highlight the relevance of complementing the scoring processes with 1) a robust and detailed interactive feedback in real-time and 2) a persistent logging subsystem to diagnose possible errors and recover from them, as well as to detect bottlenecks. We implemented a configurable logging system that records every operation in detail. For example, when a query is launched to any of the databases, the query text, the connection details, the number of records involved, and the start and end times are dumped into the log store.

Another usual administrative task deals with user permission management. Currently, two access profiles are supported: counsellor and supervisor. Counsellors can only access risk information from those students who are directly under their explicit supervision (typically between 12 and 50 students). Supervisors are special users who belong to the "Department of Student Attention and Orientation" and can access information on all students using the appropriate filters.

2) Final Users

The final users are the counsellors and the supervisors. Both profiles have been described above. Despite their differences, the can access the same type of information and do the same operations. After logging in, an overview of the students available is presented. The first information shown is a histogram of the corresponding students according to the last calculated risks (see Fig. 3). The aim is to give a quick overview of the situation in terms of risk distribution. In the histogram, each "brick" represents one student; further details about him (name and last risk score) can be accessed by hovering the cursor over it. Please note that fake personal data have been used in all the figures for the sake of privacy.

Fig. 3. Web app: Histogram of students by risk level (counsellor view).

Below the histogram, a table is shown including some details of each student, as name, study, enrollment year, last calculated risk, variation regarding previous period's risk, last time its detailed view was accessed, last time a retention action was registered or type and comments of the last retention action. In this view, some additional controls are presented to the supervisors in order to help them to deal with the huge amount of data accessible (since hundreds of students can appear in a single view):

--Filtering: allows the selection of certain subsets of students based on specific attributes (degree, counsellor, etc.).

--Selection & massive operations: allow the selection of a subset of students and the execution of actions on them (e.g., registering a massive non-personalized retention action like sending a standard welcome email, etc.).

The user can access the detailed view of a specific student by clicking either on the brick that represents him in the histogram or in the corresponding row of the list. This view consists of four panels that provide detailed information about the selected student (see Fig. 4). The content of each panel is:

--Student summary information. Name, age, gender, contact, studies, self-description (as it appears in his Moodle profile) and latest calculated risk presented in a dynamically coloured gauge bar along with the date it was calculated.

--Chart depicting the historical evolution of the student's risk scores. The current and historical values of the dropout risk calculated for the student are displayed. Time is represented in the X-axis and risk values in the Y-axis. The points represent risk scores. The chart is interactive: users can zoom and select risk scores to obtain further information, i.e. the explanation labels of the one selected.

--Score explanation panel. This panel presents a set of coloured labels linked to the risk score selected in the chart. These labels conform an explanation for the selected risk score: each of them corresponds to a single feature evaluated in the scoring process; the text on the label includes information about both the feature and the exact value for this feature in this scoring process for this student; and the colour (green/red) informs of the impact of this value on the final score: red means that it increased the risk, green means that it decreased it.

--Retention action panel. It includes all the functionality related to retention actions recording and display. It lists, chronologically, all the retention actions registered for the student: type of action (e.g., phone call, e-mail, personal

1939-1382 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2911608, IEEE Transactions on Learning Technologies

meeting, etc.), date, time, name and role of the user who registered the action, subjective evaluation (negative, neutral or positive impact, represented by three "smiley" icons), and user comments and observations. It also allows the user to edit or delete actions if he has the corresponding permissions (that is, if he is either the action owner or a supervisor). Finally, it allows the user to register a new action using a simple form.

Fig. 4. Web application – Student's detailed View.

B. *Results: How accurately have models been predicting the risk of dropout?*

After the first semester in production (2nd semester of the 2016-17 academic year), and now that we know which of the students in that semester finally dropped-out and which persisted, the natural question is "how accurately have the production models been predicting dropout risk?". The following charts were created to answer this question:

1) *Density charts*

Figs. 5a) and 5b) represent the risk distributions in subpopulations of persistent / dropped out students for the 2016-17 academic year as density charts. The final, real student behaviour is represented by colors: green for persistent students and red for dropouts. If the models perform well, the curve corresponding to non-dropouts (green) must have most of its area in the left side of the chart (low risk values) and the curve corresponding to the dropout students (red) must have most of its area in the right side of the chart (high risk values). The less both curves overlap, the better. Separate charts are created to compare the early models (at enrollment time, Fig. 5a) with the combined average performance of all the models (Fig. 5b).

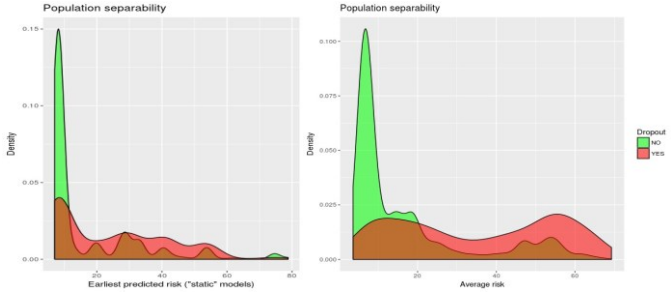


Fig. 5. Risk distributions for persistent (green) /dropout (red) students
a) Earliest risk distribution b) Average risk distribution

2) *Cumulative distribution plots*

The lines in Fig. 6a) and Fig. 6b) represent the percentage of students under a certain risk level in persistent and dropped out subpopulations of students for the 2016-17 academic year. The final, real student behavior is represented by colors: non-dropouts (green) or dropouts (red). Separate charts have been created to compare the early models (at enrollment time, Fig. 6a) with the later periodic models (Fig. 6b).

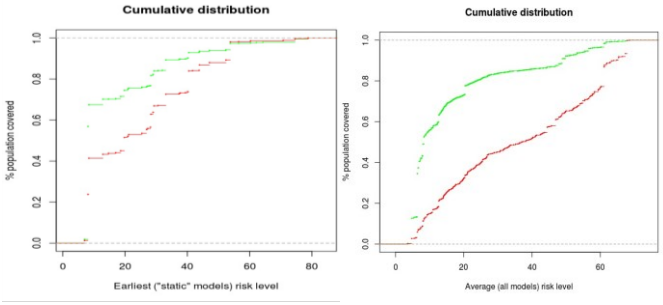


Fig. 6. Percentage of persistant/dropout students under a certain risk level
a) Early predictions (early risk level) b) All predictions (average risk level)

The conclusions after examining the previous charts follow. Evaluated as a whole, the C5.0 models used in production provided a reasonably good separability of classes, close to the performance of the Random Forest models tested during the research stage. For example, addressing all the students with an average risk of dropping out greater than 25% would have resulted in addressing a 60% of real dropouts and only a 20% of persistent students (i.e. false dropouts). As expected, this is below (but close) to the performance of RF models in the lab: at a fixed 20% of false positives, the average sensitivity obtained was 65.5%. Fig. 7 shows an example of c5.0 tree model for period 10 and recurrent students.

```
Read 10355 cases (138 attributes) from undefined.data
Read misclassification costs from undefined.costs

Decision tree:

num.view.course.delta > 17: NO (7397/330)
num.view.course.delta <= 17:
:...porc.avance.estudio > 77.08: NO (562/26)
    porc.avance.estudio <= 77.08:
    :...num.creditos.B > 25: NO (336/37)
        num.creditos.B <= 25:
        :...num.view.course.delta <= 0:
            :...num.quizs.cantidad <= 19: SI (1178/590)
            :   num.quizs.cantidad > 19: NO (218/58)
            num.view.course.delta > 0:
            :...creditos.mat.2s > 0: NO (167/23)
                creditos.mat.2s <= 0:
                :...nota_media.assign.asignaturas <= 3: SI (338/224)
                    nota_media.assign.asignaturas > 3: NO (159/30)


Evaluation on training data (10355 cases):

        Decision Tree
      ----------------
      Size      Errors   Cost

         8 1318(12.7%)   0.20   <<


       (a)   (b)    <-classified as
      ----  ----
      8335   814    (a): class NO
       504   702    (b): class SI
```

Fig. 7. Example of c5.0 tree: model for period 10 and recurrent students.

As expected (see, e.g., [31]), the static early models obtain the lowest performances. Nevertheless, they are able to separate classes well enough to be useful: addressing all the students with an early dropout risk value over 25% would have resulted in a coverage of approximately 44% of real dropouts and only

1939-1382 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2911608, IEEE Transactions on Learning Technologies

about 20% of persistent students (i.e. false dropouts). This is again below but close to the performance of the reference RR models: at a fixed 20% of false positives, they yielded an average sensitivity of 49.5%, as shown in Table I previously.

It is important to note that success may have affected the figures represented in these charts negatively: if the retention actions actually had an impact, some of the students at high risk who were actually going to drop out will eventually have been retained. This is a case of success: the students have been retained. However, they would appear as errors in this evaluation (false positives): they were labeled high-risk, but did not finally dropout.

C. *Results: Retention actions and their effect.*

At the time of writing this paper, SPA has been working for three consecutive semesters during the last two academic years: second semester of 2016-17, first semester of 2017-18 and second semester of 2017-18, and is actively being used in the current academic year. During these three semesters, the system has generated about 117,000 risk scores that have been used to assess the dropout risk of about 5,700 undergraduate students, generating about 13,000 retention actions registered in the system:

--81% of these actions correspond to written interactions (emails) while 19% of them were phone calls.

--77% of them were labeled "neutral" by the counsellors/supervisors, 22% were "positive", and the remaining 1% were "negative". The actions labelled "negative" correspond to situations in which it was not possible to contact the student by any mean, or complaints were expressed, or confirmation of dropout was received from the student.

It is worth mentioning that interactions with at-risk students reveal personal situations and stories that not only allow for a personalized approach to the student, but also provide interesting insights about the types of situations that endanger the continuity of students' learning projects. Some extracts from real comments (with fictitious names of students) taken from the annotations associated with retention actions are:

"Susan tells me that she has become pregnant and she is moving, so she's not being able to cope with her studies this semester. She hopes to take the exams in September (...)"

"Patrick is happy but says to be struggling to combine his academic and professional life properly, since he is in his first year and it's a new situation for him. He is very grateful for the call. His intention is to register for September exams (...)"

"I write an email to the student to ask why he has not registered for the July exams. He explains that he had an unexpected work trip and has been abroad for about two months (...)"

At the time of writing this paper, only those data corresponding to the first semester of usage (second semester of 2016-17) are complete. The reason is that later students (from 2017-18) cannot be labelled as dropout or persistent until March 2019. Therefore, the results presented base on the retention actions taken during that semester. They are promising and hint towards an actual positive impact:

--The population of students with a high average risk (>50%) that have persisted, have received, on average, more retention actions than those students with a high average risk that have dropped out (0.89 vs. 0.69).

--Among all the retention actions, phone calls seem to be more effective than written interactions. The population of students that have received successful phone calls present a lower dropout rate when compared to control groups that have not (18% vs. 22%).

Nevertheless, the amount of data available is still small to draw definite conclusions, something that will be addressed once the 2017-18 data becomes available.

VI. Challenges and Lessons Learnt

Turning laboratory results into a real production software system raises several challenges that need to be anticipated and effectively addressed. We have classified them in seven categories, each of them explained in detail next.

A. *Cost Effectiveness & Viability*

The benefits of successful retention initiatives are well known, to name a few:

--Improvement in academic metrics such as higher graduation rates and lower dropout rates.

--Financial profitability derived from the fact that, in general, the cost of attracting a new student is significantly higher than the cost of retaining one and, therefore, retention yields a bigger return on investment (ROI); i.e., investing in retention has a higher ROI than investing in attraction.

--Improvements in student satisfaction and institutional reputation.

However, developing, using and maintaining a dropout prevention system cost money and resources. In a real environment, cost effectiveness is not a minor issue, and, in most cases, determines viability. For this reason, the decision whether to carry out such an initiative must consider metrics related to the cost and expected value of exploiting the proposed system. We addressed this by simulating scenarios in order to assess the viability of the project through the estimation of two expected quantitative outcomes: the expected impact on dropout rates and the expected economic return on investment.

The model considered 11 parameters, divided in two sets. Six of them can be obtained or calculated from current and historical data available at the institution, such as: current dropout rates, number and status of enrolled students & credits, price-per-credit/status, etc. The other five parameters are estimated in order to produce the different scenarios and relate to costs and performance: i) increased cost of addressing one at-risk student through retention actions; ii) annual cost of maintaining and updating the retention system; iii) sensitivity of predictive models, estimated through test-set validation and cross-validation techniques; iv) false-positive rate of predictive models, also estimated through test-set validation and cross-validation techniques; and v) success rate of addressed at-risk students: percentage of students addressed that finally will not leave.

The model was implemented in an R web application and different scenarios were assessed. It is worth highlighting that,

1939-1382 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2911608, IEEE Transactions on Learning Technologies

in addition to numeric and graphical outputs, the results were also generated in natural language, to ease their interpretation and the subsequent decision-making process by the corresponding authorities. The natural language descriptions of the simulated scenarios were like this:

*"According to the specified parameters, of the **e** students enrolled, **d** will drop out if no retention actions are carried out. If the retention system is used, it will generate about **n** alerts for at-risk students, of which **a** would be accurate alerts, and **f** would be false alarms. Therefore, **r** students would be addressed through retention actions, with a total retention cost of **€ c**. As a result of these actions, **r** students would be retained and would not drop out, generating an additional income of **€ i** the following year, for which the return on investment would be **€ m** (after discounting **€ c** invested in retention actions and **€ s** invested in the retention system), and the dropout rate would decrease in **x** percentage points, yielding a **p%**".*

The decision of launching the initiative was taken once the model yielded positive results even with conservative parameter estimations, especially in the success rate of the at-risk students addressed, which was estimated as low as 5%. The lessons learned are:

--Good results can be achieved in the laboratory regarding predictive models, but may not be feasible in a real production system due to sustainability issues. This must be evaluated before launching the project.

--There is usually enough information to simulate reasonably accurate parameterized scenarios to help with decision-making processes.

B. Changing Organizational/Operational Context

In the initial stages of the project, the focus was on demonstrating the viability of generating the expected results regarding predictions, but once it was clear that they could be got, the focus shifted to another relevant question: who will use the results and how?

The initial answer to this question was based on a decentralized counseling model: the main users and consumers of the information about dropout risk would be the counsellors. The web application interface was designed assuming that a big number of users in the application would manage a small number of students.

However, some organizational changes took place soon: a new central "Department of Student Attention and Orientation" was created. From then on, they would be the main users of the system and, therefore, the user model drastically changed: there would be a small number of users in the application, each one in charge of a big number of students.

The interfaces, initially designed to display small numbers of students, had to be modified to support, furthermore, the display of several hundred: ordering and filtering functionality had to be expanded and added; the possibility of registering massive actions had to be included to support new one-to-many operations; etc. The lessons learned are:

--The question of "who will use the system and how?" must necessarily follow and is as important as the question of "can we produce the expected results?". Lab work usually ends with the latter, but production one requires investing time in answering the former.

--Expect changes soon after rollout. Usually, reality does not fit the expected usage scenarios perfectly, and quick, agile changes may be needed. This is especially true if the organizational structure supporting retention is young in the institution or is being created at the same time as the retention system.

C. Model explainability

As stated in [13], "*one of the main drawbacks of the methods commonly used in data mining is that they are difficult to interpret because they act as black boxes, providing results without explanation*".

It is well known that black-box modelling techniques tend to provide better predictive performance than white-box ones. Since modern techniques like convolutional neural networks were demonstrating very promising results in other domains, their use in our project during the research stage was a very tempting idea. However, the fact that the goal was to build a real system and not a lab product, forced us to choose white-box techniques. The reason is clear: in practice, model explainability is an essential requirement: every prediction must be individually explained, due to the two facts explained below:

1) Credibility and user adoption.

Users will not give credibility to predictions they do not understand. Even with a white-box approach, much of the issues registered while using the system relate to demands of explanation. The most frequent were:

--The reason why the risk level calculated by the system did not match the intuition of the corresponding counsellor, or the cause-effect relationship is unclear.

--The reason why there was an asymmetric impact of a certain feature in the risk value. For example, in a certain model, if the value of one of the features for a student is under a certain threshold, the risk increases by +x%, but if the value is over that threshold, the risk usually does not decrease by -x%, as users tend to expect.

It is paramount to be prepared to address these issues quickly and effectively, for two reasons:

--User adoption can be severely compromised if users do not get timely and satisfactory answers to these questions.

--The reasons given by users when they suspect that the system might be giving a wrong risk level are an invaluable source of expert knowledge, as they often point to ideas that are key to correcting, refining and improving the underlying predictive models.

2) Preventing the effect by understanding the causes

Knowing the logic used by the models to yield the dropout risk levels gives the possibility of hinting valuable root causes for desertion, and, in turn, gives the chance to develop tailored and more effective retention actions. For example, in our initial models, the level of completion of the Moodle user profile (photograph, country/city, self-description, etc.) systematically appeared as an important variable for prediction, linking poor or empty user profiles to dropout, and rich user profiles to persistence. There may be several explanations for this, like:

1939-1382 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2911608, IEEE Transactions on Learning Technologies

--Students who do not complete their profile do so because they are not familiarized with on-line web applications and probably do not even know such option exists, and the proper use of the LMS may be a challenge for them.

--Students who do not complete their profile do so because they present certain personality traits that are not well suited to distance education.

--Students who do not complete their profile do so because they do not feel engaged or integrated in a bigger, live community, but see themselves as isolated students.

Without evaluating the validity of each of these potential explanations, it is clear that the fact of knowing what pieces of information the models are using to increase/decrease the dropout risk levels enables the elaboration of hypotheses for the possible root causes and the design of tailored and more effective retention actions.

Communicating the subjacent predictive models' logic to users is not trivial and represents a challenge. We have found that the usual model logic representations used in labs (for example, depiction of decision trees) are neither well received nor understood by the users. In an effort to communicate this valuable information better, we created an infographic that summarizes the most prevalent logic of the models in a more user-friendly format (see Fig. 8). The lessons learned are:

--Model explainability is an essential requirement in a real-world application.

--Even with explainable white-box models, many requests will be made and a great communication effort will be needed to explain the logic of the model to the users. This effort pays off in terms of user adoption, high value feedback and model improvement.

Fig. 8. Infographic to communicate the risk-aggravating factors to the users.

D. Evolving Systems Integration

Lab conditions are usually ideal in two key aspects. Firstly, they are free from outside-world constraints, as the developed artifacts do not have to be integrated anywhere outside their own research context. Secondly, their technical context does not change during research. The conditions stay the same. Data, software versions, database schemas, etc. are usually static. This is not true for real-world systems for the following reasons. A proper integration with the existing environment is required and there may be non-trivial conflicts between lab conditions and IT corporate requirements (technology stack restrictions, authentication mechanisms, security/connectivity restrictions, etc.). In addition, technical changes happens all the time. Keeping on par with technical changes happening in the surrounding IT environment is a very time consuming but essential task.

Our experience was the following. On one hand, due to security reasons, we had to adapt to some changes in the versions of many of the components of the technology stack (databases and application server). This forced us to re-test the architecture with the new configuration and versions. On the other hand, the constant evolution of the source systems, especially the LMS, forces us to periodically rewrite and re-test several parts of the system. For example, at the time of writing this paper we are adapting the ETL and model-generation modules to deep changes in the internals of the Moodle messaging system included in the last version. Last year we had to adapt to deep changes in the internals of the Moodle logging system, etc. These changes pose a double risk to the system, since sometimes the changes require not only technical adaptations, but also semantic adjustments. For example, changes in the granularity of the new Moodle logging system implied the development of complex logic to harmonize historical logging data with new logging data to generate correct and time-consistent features to feed model training and scoring. The lessons learnt are these:

--Expect changes when implementing your lab-developed system in the real world due to IT corporate requirements.

--Plan for a constant and sustained technical adaptation effort throughout the whole lifecycle of the system. This is, in our experience, the most time-consuming, critical and costly recurring task, and part of the maintenance effort described before.

E. Model validity

The validity of predictive models relies on one hard assumption: the immutability of the context where the models were trained and the predictions are being made. Unfortunately, reality is hardly ever this way, and this is another reason of why many lab-generated models' applicability is severely compromised: lab conditions simply do not exist anymore when results are published, and hence models are no longer valid.

In the previous section, we referred to the technical changes that threaten the applicability of the models. In this section, we will refer to two other types of changes, subtler and hence posing a bigger risk to model validity, often requiring an in-depth review and a complete reconstruction of the models.

1) Administrative changes

Administrative institutional processes change with time: changes in legislation, changes in the internal regulations, adaptive changes to accommodate new administrative situations, or simply improvements in the way certain situations are handled and registered in the corresponding system.

1939-1382 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2911608, IEEE Transactions on Learning Technologies

This kind of changes pose a bigger risk for the integrity of the system because they may go unnoticed, causing the system to run smoothly but flawed. For example, the way in which certain fee discounts are granted and recorded in the academic system changed recently. If this change had gone unnoticed, the system would have (incorrectly) assumed that no students were granted the discount in the last academic year. Since there is a feature that explicitly checks this situation, the models would have yielded flawed risk values.

2) Methodological changes.

The other kind of changes that may have an impact on the system are methodological or pedagogical ones The way in which certain teaching activities are handled may change, internal academic rules may change, or even worse, legitimate activities that effectively "contaminate" certain features may occur. Again, the biggest risk of this type of changes is that they very well may go unnoticed. A real example using a feature mentioned in a previous section: starting from a certain academic period, the percentage of students with "rich" Moodle user profiles started to grow abruptly. This change was detected, and after investigating, we discovered that, from a certain moment, as part of a compulsory introductory course, every student had to complete his profile, disabling any previous predictive power of this feature. As a result, this feature had to be excluded from the subsequent models.

3) Mitigating the risks.

To lower this risk of being affected by unnoticed methodological or administrative changes, two mitigating actions must be performed at the beginning of each new academic cycle:

--The pre-calculation and statistical review of each feature, in order to detect changes in the distributions of the data that may reveal underlying changes in the processes or administrative realities they represent.

--The generation and periodic review of a checklist containing the administrative processes that, if changed, may influence the system's interpretation.

F. System maintenance and evolution

As in any software system in production, maintenance is necessary in its various forms: i) corrective, that is, solving bugs and malfunction detected in the system during its operations; ii) adaptative, i.e., adjusting the system to fit the changing environment in which it operates; iii) evolutive, functionally and technically, implementing improvements and expansions, assuring that the system does not decay into technical obsolescence. Supporting maintenance requires at least two actions:

--Defining and communicating a SLA (Service Level Agreement), so users know where to go when problems and new needs arise and what to expect.

--Setting up and managing an Issue Tracking System (or integrating with the existing corporate one, as we did).

The main lesson learned is that delivering a proper maintenance is essential for mid and long-term user adoption, and requires setting up and allocating permanent resources and a relevant, sustained effort. This, in fact, turns the project into a service. In our experience, failing to foresee this and persisting in managing the effort with a "project" mindset is one of the biggest threats when trying to move from the lab to production.

G. Legal compliance

During the lab stage, the use of anonymized data is the common approach. However, in production systems, the required legal rights of the data subjects (in this case, the students) must be supported and enforced, especially when personal information is involved. This means that, in addition to the typical data privacy and authorization mechanisms, specific functionalities had to be implemented to support these other rights, in our case granted by the legislation of the EU:

--"Information and Access" right: functionality to export all the information contained in the system about any individual in a machine-readable format.

--"Erasure right": functionality to remove all the data about any individual from the system.

--"Restriction of processing" right: functionality to stop the processing of any individual's data by the system and the consequent cessation of derived information production.

The lesson learned is that special attention must be paid to the legal implications of the system, studying the regulations that may apply and implementing all the functionalities needed to support the corresponding rights of the data subjects.

VII. Conclusions

In this article, we have presented the work done to support dropout risk prevention in a real online e-learning environment: a Spanish distance university with thousands of undergraduate students. The main goal is to prevent students from abandoning the university by means of retention actions oriented to those at risk of dropout, trying to guarantee the effectiveness of institutional efforts in this direction.

With this goal, firstly we did lab research and simulated realistic scenarios in order to assess the viability of the solution proposed. Once its feasibility was clear, we generated predictive models based on the C5.0 algorithm and developed SPA, an EWS that uses these models to generate student dropout-risk predictions and registers the resulting retention-oriented interventions. Both early predictions and updated periodic ones are supported, considering new and recurrent students separately.

Data from more than 11,000 undergraduate students have been used as training samples, with more than 120 features describing each of them (either obtained from the academic and learning management systems or calculated). About 117,000 risk scores have been computed to predict the dropout risk of about 5,700 students and around 13,000 retention actions have been recorded.

The models used in production provided a reasonably good separability of classes, close to the performance of the Random Forest models tested in the lab: addressing all the students with dropout risk scores greater than 25% would have covered 60% of real dropouts and 20% of persistent students (and it must be taken into account that some of these persistent students could have been dropouts if no interventions had occurred).

1939-1382 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2911608, IEEE Transactions on Learning Technologies

Turning the laboratory results into a real production system raises challenges that need to be effectively addressed for success. They deal, mainly, with estimating cost effectiveness and viability beforehand, addressing changes in organizational and operational contexts, supporting model explainability, integrating evolving systems, creating valid models while mitigating risks, supporting and enforcing legal rights, and guaranteeing a corrective, adaptive and evolutive maintenance. We learnt many lessons while putting the system into production, in summary:

--Good results achieved in the laboratory may not be possible in the long term in a real production system due to sustainability issues. It is appropriate and feasible to simulate reasonably accurate scenarios to help with decision-making processes.

--Moreover, the context evolves and lab conditions might not exist anymore, making the models invalid for production.

--Knowing who will use the system and how it will be used is essential to design proper interfaces and processes.

--Technical changes will be needed soon in the production context and should be addressed quickly and with agility.

--Many changes may be needed due to IT corporate requirements when switching from lab to production.

--Model explainability is essential in real-world predictive systems: the prediction logic must be clearly explained to the users at student level for a good system adoption as well as to receive useful feedback from them.

--Legal regulations must be considered and user rights must be supported properly.

--It is necessary to make a constant and sustained technical adaptation effort throughout the whole lifecycle. It will be time-consuming, critical and costly, but it will be worth it.

It is worth highlighting that this work has been possible because the university managers had the vision, years ago, of storing all the data, anticipating its future value. We thank the board of UDIMA for giving us permission to publish this article with the restriction of not disclosing any personal information.

As future work, we will contrast the results obtained with those from the analysis of full data from 2017-18 (when the enrollment data for 2018/19 is fully available, so that we can know which students persisted and which ones finally dropped out) in order to continue evaluating the real-world performance of the prediction models developed and the effectiveness of the retention actions taken. We hope that this experience, including the lessons learnt while putting SPA into production, can be useful for the community when developing strategies for improving retention elsewhere.

REFERENCES

[1] R. Conijn, C. Snijders, A. Kleingeld, and U. Matzat, "Predicting Student Performance from LMS Data: A Comparison of 17 Blended Courses Using Moodle LMS," IEEE Trans. on Learning Technologies, vol. 10, no. 1, pp. 17-29, March, 2017, DOI: 10.1109/TLT.2016.2616312.

[2] A. Jokhan, B. Sharma, and S. Singh, "Early warning system as a predictor for student performance in higher education blended courses," Studies in Higher Education. DOI: 10.1080/03075079.2018.1466872. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/03075079.2018.1466872.

[3] J.A. Lara, D. Lizcano, M.A. Martínez, J. Pazos, and T. Riera, "A system for knowledge discovery in e-learning environments within the European Higher Education Area – Application to student data from Open University of Madrid, UDIMA," Computers & Education, vol. 72, pp. 23-36, March 2014, DOI: 10.1016/j.compedu.2013.10.009.

[4] J. Daniel, "Making Sense of MOOCs: Musings in a Maze of Myth, Paradox and Possibility," J. of Interactive Media in Education, vol. 2012, no. 3, 2012, DOI: 10.5334/2012-18.

[5] UNED, "Facts and data about UNED," UNED Data, Sep. 10, 2018. [Online]. Available: http://portal.uned.es/portal/page?_pageid=93,24305391&_dad=portal&_schema=PORTAL

[6] The Open University, "Annual report of the Open University", The Open University Data, Sep. 10, 2018. [Online]. Available: http://www.open.ac.uk/about/main/sites/www.open.ac.uk.about.main/files/files/Annual-report-2017-18_2.pdf

[7] Ministry of Education of Spain, "Data of Spanish universities of 2015-2016 academic course," Sep. 2018. [Online]. Available: https://www.mecd.gob.es/dms/mecd/servicios-al-ciudadano-mecd/estadisticas/educacion/universitaria/datos-cifras/datos-y-cifras-SUE-2015-16-web-.pdf

[8] R. Baker, and K. Yacef, "The state of educational data mining in 2009: A review and future visions," J. Educ. Data Mining, vol. 1, no. 1, pp. 3-17, Oct. 2009.

[9] C. Romero and S. Ventura, "Educational Data Mining: A Review of the State of the Art," IEEE Trans. Systems, Man and Cybernetics, Part C: Applications and Reviews, vol. 40, no. 6, pp. 601-618, Nov. 2010, DOI: 10.1109/TSMCC.2010.2053532.

[10] G. Kostopoulos, S. Kotsiantis, O. Ragos, and T.N. Grapsa, "Early Dropout Prediction in Distance Higher Education Using Active Learning," in Proc. IISA, Larnaca, Cyprus, 2017, pp. 1-6.

[11] E. Howard, M. Meehan, and A. Parnell, "Contrasting Prediction Methods for Early Warning Systems at Undergraduate Level," The Internet and Higher Education, vol. 37, pp. 66-75, Apr. 2018, DOI: 10.1016/j.iheduc.2018.02.001.

[12] J.M. Ortiz-Lozano, A. Rua-Vieites, P. Bilbao-Calabuig, and M. Casadesus-Fa, "University student retention: Best time and data to identify undergraduate students at risk of dropout," Innovations in Education and Teaching International, to be published. DOI: 10.1080/14703297.2018.1502090.

[13] C. Lacave, A.I. Molina, and J.A. Cruz-Lemus, "Learning Analytics to identify dropout factors of Computer Science studies through Bayesian networks," Behavior & Inf. Tech., vol. 37, no. 10-11, pp. 993-1007, Jun. 2018. DOI: 10.1080/0144929X.2018.1485053.

[14] J.B. Heppen and S.B. Therriault, "Developing Early Warning Systems to Identify Potential High School Dropouts," National High School Center, Washington D.C., USA, Jul. 2008. [Online]. Available: http://files.eric.ed.gov/fulltext/ED521558.pdf

[15] M.A. Mac Iver and D.J. Mac Iver, "Beyond the Indicators: An Integrated School-Level Approach to Dropout Prevention," Center for Equity and Excellence in Education, Arlington, VA, USA, Aug. 2009. [Online]. Available: https://files.eric.ed.gov/fulltext/ED539776.pdf

[16] H.P. Beck and W.D. Davidson, "Establishing an Early Warning System: Predicting Low Grades in College Students from Survey of Academic Orientations Scores," Research in Higher Education, vol. 42, no. 6, pp. 709-723, Dec. 2001.

[17] J.E. Knowles, "Of Needles and Haystacks: Building and Accurate Statewide Dropout Early Warning System in Wisconsin," J. Educ. Data Mining, vol. 7, no. 3, pp. 18-67, Jul. 2015.

[18] L. Najdi and B. Er-Raha, "A Novel Predictive Modeling System to Analyze Students at Risk of Academic Failure," Int. J. of Comp. App., vol. 156, no. 6, pp. 25-30, Dec. 2016.

[19] M. Adelman, F. Haimovich, A. Ham, and E. Vazquez, "Predicting school dropout with administrative data: new evidence from Guatemala and Honduras," Education Economics, vol. 26, no. 4, pp. 356-372, Feb. 2018. DOI: 10.1080/09645292.2018.1433127.

[20] S. B. Shum and R. D. Crick, "Learning dispositions and transferable competencies:

1939-1382 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2911608, IEEE Transactions on Learning Technologies

Pedagogy, modelling and learning analytics," in Proc. LAK, Vancouver, Canada, 2012, pp. 92–101.

[21] D. T. Tempelaar, B. Rienties, and B. Giesbers, "In search for the most informative data for feedback generation: Learning analytics in a data-rich context," Comp.. Human Behavior, vol. 47, pp. 157– 167, Jun. 2015.

[22] C. Marquez-Vera, A. Cano, C. Romero, A.Y.M. Noaman, H.M. Fardoun, and S. Ventura, "Early dropout prediction using data mining: a case study with high school students", Expert Systems, vol. 33, no. 1, pp. 107-124, Feb. 2016, DOI: 10.1111/exsy.12135.

[23] G. Kostopoulos, S. Kotsiantis, and P. Pintelas,, "Predicting Student Performance in Distance Higher Education Using Semi-supervised Techniques," in Proc. MEDI, Rhodes, Greece, 2015, pp. 259-270.

[24] S. Brooman and S. Darwent, "Measuring the beginning: a quantitative study of the transition to higher education", Studies in Higher Education, vol. 39, no. 9, pp. 1523-1541, Oct. 2014.

[25] M. O'Cummings and S.B. Therriault, "From Accountability to Prevention: Early Warning Systems Put Data to Work for Struggling Students", Early Warning Systems in Education, Washington, D.C., USA, May. 2015. [Online]. Available: https://files.eric.ed.gov/fulltext/ED576665.pdf

[26] OCU. UNIVERSITAS XXI – ACADÉMICO. [Online]. Available: http://www.ocu.es/productos/universitas-xxi-academico/

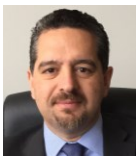
[27] MOODLE. The Moodle project. [Online]. Available: https://moodle.org/

[28] N-B. Şara, R. Halland, C. Igel, and S. Alstrup, "High-school dropout prediction using machine learning: a Danish large-scale study," in Proc. ESANN, Bruges, Belgium, 2015, pp. 319-324.

[29] A.J. Bowers, R. Sprott, and S. Taff, "Do We Know Who Will Drop Out? A Review of the Predictors of Dropping out of High School: Precision, Sensitivity and Specificity," The High School Journal, vol. 96, no. 2, pp. 77-100, Dec. 2013, DOI: 10.1353/hsj.2013.0000.

[30] L. Kennelly and M. Monrad, "Approaches to Dropout Prevention: Heeding Early Warning Signs with Appropriate Interventions," National High School Center, Washington, D.C., USA, Oct. 2007. [Online]. Available: https://files.eric.ed.gov/fulltext/ED499009.pdf

[31] M.N. Mustafa, L. Chowdhury, and Md.S. Kamal, "Students Dropout Prediction for Intelligent System from Tertiary Level in Developing Country," in Proc. ICIEV, Dhaka, Bangladesh, 2012, pp. 113-118.



Alvaro Ortigosa was born in San Carlos de Bariloche, Argentina, in 1968. He holds a Ph.D. on Computer Science from the Universidad Autónoma de Madrid in 2000, a M.S. on Computer Science from the Universidade Federal de Rio Grande do Sul in 1995 and a degree on Computer Science from the Universidad Nacional del Centro de la Prov. de Buenos Aires in 1993. He is director of the Research Institute for Forensics and Security Science of UAM since 2017 and Associate Professor at the Department of Computer Science of UAM since 2001. His main research lines are adaptive systems and user modeling, application of datamining for user model acquisition, personality and emotion detection through text and virtual social network analysis, and application of datamining to risk analysis. He has (co)authored more than 60 papers in international journals and conferences. Mr. Ortigosa is member of European Cybercrime Training and Education Group.

Rosa M. Carro was born in Madrid, Spain in 1975. She holds a Ph.D. and a B.S. degree on Computer Science from Universidad Autónoma de Madrid in 2001 and 1997, respectively. She is Professor of the Department of Computer Science at UAM. She visited the Universidade de Aveiro (2001-2002) and the Technische Universität München (2002-2003) to do research on adaptive educational games and collaborative adaptive e-learning systems. Currently she is interested in automatic user model acquisition, sentiment and social network analysis, adaptive e-learning systems, attention to diversity, educational data mining and learning analytics. She has co-organised several international workshops, edited special issues in international journals and coauthored around 100 papers in her main research areas. Mrs. Carro is member of the director board of the Association for the Development of Educational Informatics (ADIE).

Javier Bravo-Agapito was born in Salamanca, Spain in 1978. He holds a B.S. degree in Statistics from the UCM, Madrid (Spain) in 2000, a B.S degree in Computer Science from UAM in 2004, and a Ph.D. in Computer Science and Telecommunications from UAM in 2010. From 2005-2009, he was Research Assistant with the GHIA group, UAM. In 2005, he got a four-year research grant from the Spanish Ministry of Education and Science. Since 2010, he is Assoc. Professor and Senior Researcher at the Madrid Open University (UDIMA), Spain. He has collaborated with Prof. Serge Garlatti at Télécom Bretagne, Brest, France, and with Prof. Peter Brusilovsky at the University of Pittsburgh, Pennsylvania. Currently, his research interests focus on e-learning, adaptive educational hypermedia systems, and data mining. He was member of the American Association for the Advancement of Science, and published relevant articles in IEEE Communications and IEEE Access.

David Lizcano was born in Madrid, Spain in 1983. He holds a Ph.D. in Computer Science (2010), and a M.Sc. degree in Research in Complex Software Development (2008) both from UPM. He got a research grant from the European Social Fund under their Research Personnel Training program, the Extraordinary Graduation Prize for best academic record of 2005 graduates, the 2005 Academic Attainment Prize of UPM General Foundation, and the 2006 National Accenture Prize for the Best Final-Year Computing Project. Since 2011, he is Professor and Senior Researcher at the Madrid Open University (UDIMA), where he is currently Vice-rector of Research and Doctorate. He is involved in several national and European funded projects related to EUP, Web Engineering, Enterprise 2.0 technologies, Paradigms of Programming and Human-Computer Interaction. He has published more than 25 papers in prestigious international journals, attended more than 70 international conferences and participated in more than 10 international EU R&D projects.

Juan Jesús Alcolea was born in Madrid, Spain in 1973. He studied all the courses of the Degree in Computer Science in the Technical University of Madrid (UPM), and received a M.S. degree in International Higher Education Management from the Universidad de Alcalá, Madrid, Spain, in 2010. Since 1997, he works in the field of business intelligence and data analysis in the telecom and finance sectors, focusing on Higher Education since 2002. He is co-founder and Director of Analytics in DIMETRICAL (an EdTech startup) and collaborates with online and face-to-face universities teaching big data courses as an external teacher. He was the director and co-author of the "White book of Institutional Intelligence in Universities" (2013), and co-founder of the Business Intelligence task force inside the European organization EUNIS (European University Information Systems).

Óscar Blanco was born in Madrid, Spain in 1973. He received his M.S. degree in computer Science from Technical University of Madrid (UPM), Spain in 2009. He received the PMP Certification from PMI in 2014. He is Business Intelligence Director in DIMETRICAL - an EdTech startup based in Madrid (Spain). For more than 20 years he has designed and constructed analytical solutions for information exploitation in sectors such as semiconductor manufacturing (Lucent Tech. Microelectronics), telecommunications (British Telecom) and education (University Cooperation Office). Since 2001 he combines his experience in business intelligence with university management, collaborating in the implementation of institutional intelligence solutions for more than 15 institutions in the field of Higher Education (university and government), in Spain and Latin America. Mr. Blanco is member of the Professional College of Computer Engineers of Community of Madrid.

1939-1382 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.