IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 24, 2021, accepted September 22, 2021, date of publication September 27, 2021, date of current version October 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3115851

A Real-Life Machine Learning Experience for Predicting University Dropout at Different Stages Using Academic Data

ANTONIO JESÚS FERNÁNDEZ-GARCÍA[1], JUAN CARLOS PRECIADO[2], FRAN MELCHOR[2], ROBERTO RODRIGUEZ-ECHEVERRIA[2], JOSÉ MARÍA CONEJERO[2], AND FERNANDO SÁNCHEZ-FIGUEROA[2]

[1]Escuela Superior de Ingeniería y Tecnología, Universidad Internacional de la Rioja, 26006 Logroño, La Rioja, Spain
[2]Quercus Research Group, University of Extremadura, 01003 Cáceres, Spain

Corresponding author: Antonio Jesús Fernández-García (antoniojesus.fernandez@unir.net)

This work was supported in part by the Ministerio de Ciencia, Innovación y Universidades (MCIU), Agencia Estatal de Investigación (AEI), and European Regional Development Fund (ERDF), under Project RTI2018-098652-B-I00; and in part by the European Regional Development Fund (ERDF) and Junta de Extremadura under Project IB16055, Project IB18034, and Project GR18112.

ABSTRACT High levels of school dropout are a major burden on the educational and professional development of a country's inhabitants. A country's prosperity depends, among other factors, on its ability to produce higher education graduates capable of moving a country forward. To alleviate the dropout problem, more and more institutions are turning to the possibilities that artificial intelligence can provide to predict dropout as early as possible. The difficulty of accessing personal data and privacy issues that it entails force the institutions to rely on the Academic Data of their students to create accurate and reliable predictive systems. This work focuses on creating the best possible predictive model based solely on academic data, and accordingly, its capacity to infer knowledge must be maximised. Thus, *Feature Engineering* and *Instance Engineering* techniques such as dealing with redundancy, significance of the features, correlation, cardinality features, missing values, creation or elimination of features, data fusion, removal of unuseful instances, binning, resampling, normalisation, or encoding are applied in detail before the construction of well-known models such as *Gradient Boosting*, *Random Forest*, and *Support Vector Machine* along with an *Ensemble* of them at different stages: prior to enrolment, at the end of the first semester, at the end of the second semester, at the end of the third semester, and at the end of the fourth semester. Through the construction of these predictive models that serve as inputs to a decision support system, the application of effective dropout prevention policies can be applied.

INDEX TERMS Machine learning, feature engineering, instance engineering, ensemble models, real experiences, student dropout.

I. INTRODUCTION

One out of three Spanish students leave college without completing the degree in which they enrolled, which places Spain among the countries that take least advantage of the public and private higher education due to the high dropout rates of students [1]. Academic institutions are not unaware of this problem and in recent years the possibilities that Artificial Intelligence offers are being explored to create predictive models that enable them to anticipate, prevent, and, where necessary, take action to alleviate student dropout rates.

Different approaches exist when machine learning is used to build predictive models that anticipate student dropout according to the type of data used to create them. Some of them make use of personal data about the students themselves, their families, and their environment; others benefit from the interaction data stored by online learning environments or virtual classroom tools to analyse student behaviour and interaction with the teaching material; and finally, there are occasions in which institutions only have access to academic data without relevant personal information, which is

The associate editor coordinating the review of this manuscript and approving it for publication was Haiyong Zheng.

to say strictly the administrative information needed from the student.

In this paper, we focus our research efforts on the last case, when only basic information about the students is available for analysis. Although better results are likely to be achieved with the availability of every possible type of data, our interest is to get the best out of these basic data because every institution has access to them. This way, we avoid two major problems or inconveniences: a) dealing with external tools where it is necessary to gather data and consolidate them, requiring extra programming efforts which involve processes that cannot always be easily implemented by institutions; and b) the use of personal data, which is only provided for specific, legitimate, and necessary purposes of implementing appropriate security protocols. These follow European directives which involve very time-consuming processes that make it difficult to deal with students' personal data.

In view of the limited data available to model and deploy a trustworthy predictive system, this paper focuses its efforts on making the most of the existing data and their capability of inferring knowledge. This implies working hard on data preprocessing, *Feature Engineering*, *Instance Engineering*, and model creation, as well as proper validation in order to obtain an accurate and reliable predictive model to serve as a decision-making system.

We are aware of the importance of detecting the possible dropout of a student as early as possible. This leads us to the creation of various models for the different stages of the students, such as before starting university and after that, at every major step in their evolution. Therefore, as previously stated, different models are created for each of these stages in order to have the best possible model at each moment. In this regard, it highlights the risk of a student dropping out at specific points in time meaning avoidance strategies can be put in place to stop this happening.

To be more precise, five different models are created, covering the following stages: 1) Prior to starting college; 2) At the end of the first semester; 3) At the end of the second semester; 4) At the end of the third semester; 5) At the end of the fourth semester. It is necessary to point out that when we refer to the semesters, we mean the semester the student is in and not how the degrees are organised.

The data feeding the models associated with each stage is optimised by applying different data preprocessing techniques that prepare the data structure for the input of the algorithms, such as encoding or normalisation, among others. Moreover, ensemble models are built where the prediction confidence of each individual model is used to build an ensemble that allows the prediction ratios to be improved and to obtain more reliable and robust predictions. Additionally, these models are connected in cascade in such a way that the output of each model serves as the input to subsequent models, meaning that the powerful influence of the knowledge generated by previous models will be useful in the subsequent stages.

The main contribution of this paper therefore lies in the possibility of determining the students' dropout probability at different key stages of higher education starting from the earliest part before university enrolment up to the fourth semester, through different interconnected predictive models. From a scientific point of view, the novelty lies in developing several independent models using slightly different data sources, where each model corresponds to a specific stage of the students' academic life. These models are connected in cascade, which allows subsequent models to be improved by feeding them with the knowledge generated by the model at an earlier stage, thus increasing the accuracy at each stage without neglecting the potential prediction at the earliest possible stage. A real-life experience is depicted in the paper using data from engineering students at a Spanish public university.

The rest of this paper is organised as follows. Section II reviews some related projects focusing on student dropout that involves working with predictive models. Section III describes a bigger picture of the work carried out to create the predictive model for dropout and summarises the proposed methodology. Section IV provides in-depth details about how the dataset was created. The main topics discussed in this section are: a) describing data sources and fetching data; b) applying *Feature Engineering* techniques to transform features with better representation; c) applying *Instance Engineering* to carefully select and process the available instances; and d) applying some final processing techniques required before the creation of predictive models. Section V builds the predictive models and analyses the experimental results through relevant metrics. Finally, the conclusions and further considerations are summarised in Section VI.

II. RELATED WORK

Increasing computing capacity and the growing ease of accessing data are encouraging the application of intelligent systems in the education sector. This section comprises an analysis and review of the most important published work focusing on student dropout prediction in higher education.

As mentioned in the introduction, the main data used to create these predictive models are: students' interaction data with online learning environments; their personal data, their family and environment; and strictly academic data generated and managed by the universities themselves. This review details works which fall mainly into these categories, and some that may not be included, purely regarding this categorisation, are mentioned as well. However, special attention is paid to the cases in which strictly academic data are used since these can be compared through metrics to our work as they share a common basis.

Regarding students interaction with Learning Management Systems (LMS), in [2], the authors make use of some features directly extracted from the LMS such as number of messages read and sent, number of discussions read or participated in, test and exercises completed and graded, and time spent on assignments, among others, to create a predictive model for

TABLE 1. Summary of related work.

Research Work	LMS	Personal Data	Academic Data	Prediction Timing
Macfadyen et al. (2010) [2]	✓			Enrolment and First Year
Marbouti et al. (2016) [3]	✓		✓	Week 5 of the first semester
Gray and Perkins (2019) [4]	✓		✓	From week 3 to week 12 of the first year
Miguéis et al. (2018) [5]		✓	✓	At the end of the first year
Chen et al. (2020) [6]	✓	✓	✓	From month 1 to month 6 of the first semester
Helal et al. (2018) [7]	✓	✓	✓	Enrolment and First Year
Mengash (2020) [9]			✓	Enrolment
Hoffait and Schyns (2017) [8]			✓	Enrolment
Fernández-García et al. (2020) [10]			✓	At the end of the first year
(Our proposal)			✓	Enrolment and at the end of each of the first 4 semesters

classification using the first two years of a dataset from the University of British Columbia (Canada). In [3], the authors make use of academic data along with the results of tests and the participation in activities obtained from the LMS of a Midwestern U.S. University to create a predictive model that starts working from week 5 of the first semester.

In [4], the authors go beyond LMS, and they have implemented a physical infrastructure (which can be seen as intrusive) that monitors students' IDs when entering classrooms or having meetings with lecturers or professors. Based on this data from Bangor University (UK), various predictive models at different stages from 3 weeks to 12 weeks during the fall semester are created to identify students with dropout risk.

There are several works that makes use of personal data. In [5], the authors create classification models using data from an engineering and technology school that belong to a European public research University. The predictive models go from just before enrolment to 10 semesters. Personal data such as their parents' jobs or education level are used. There are also cases that make use of all types of data discussed including academic data, as in works [6] and [7]. In the former, the authors use a dataset from the College of Distance Learning at Xi'an Jiaotong University (XJTUDLC) in China with private data such as marital status or ethnicity along with data extracted from the LMS, combined with academic data to create predictive models at different stages of a semester from 1 to 6 months. In the latter, the authors make use of private data from Australian University students such as parents' education or status combined with data extracted from LMS, such as resources viewed, the number of forum posts read or written, quiz attempts and finally academic data to create predictive models at two stages: at the time of enrolment and after the first year of a degree.

Regarding the predictive models created using solely academic data, which is the case most similar to ours, in [8], the authors make use of academic data, including scholarships, to make a prediction at enrolment time in Belgian Universities. In [9], the authors make use of data from the Computer Science and Information Faculty at a Saudi public university at enrolment time using pre-admission criteria such as high school grade average, scholastic achievement admission test score, and general aptitude test score, among others.

Beyond enrolment time, after one semester of college, we ourselves have worked on a decision support system with the challenge of addressing the construction of a reliable Recommender System on the basis of data which are both sparse and few in quantity, as well as being imbalanced, thus hampering the anticipatation of students' academic achievement [10]. We have also studied the importance of Feature Selection methods in Academic Data to create easy-to-explain predictive models for shorter periods of time, reducing overfitting, and avoiding the sparsity of data which these kind of datasets usually possess [11].

Table 1 displays both similarities and differences between our approach and the related works described herein regarding such aspects as:

a) Use of Data from online Learning Management Systems (LMS);
b) Use of Personal Data;
c) Use of Academic Data;
d) Time at which the prediction is made.

As shown, our approach defines several models from enrolment up to the fourth semester using exclusively Academic Data so we ensure that every university can reproduce this work (in their own particular way). In addition, our approach takes into account the output of previous stage models, meaning the whole system at each step exploits the previous knowledge generated.

III. OVERVIEW

The steps we followed to create the predictive model to serve as a decision support system are illustrated in Figure 1. With these steps, their purpose is not to define a formal methodology. Instead, they represent the guidelines on which our approach is based, and we shall describe them in detail in the course of this manuscript in order to facilitate replication of the work and encourage its implementation.

As can be seen in Figure 1, our proposal goes from data collection to the creation of the models and consists of six

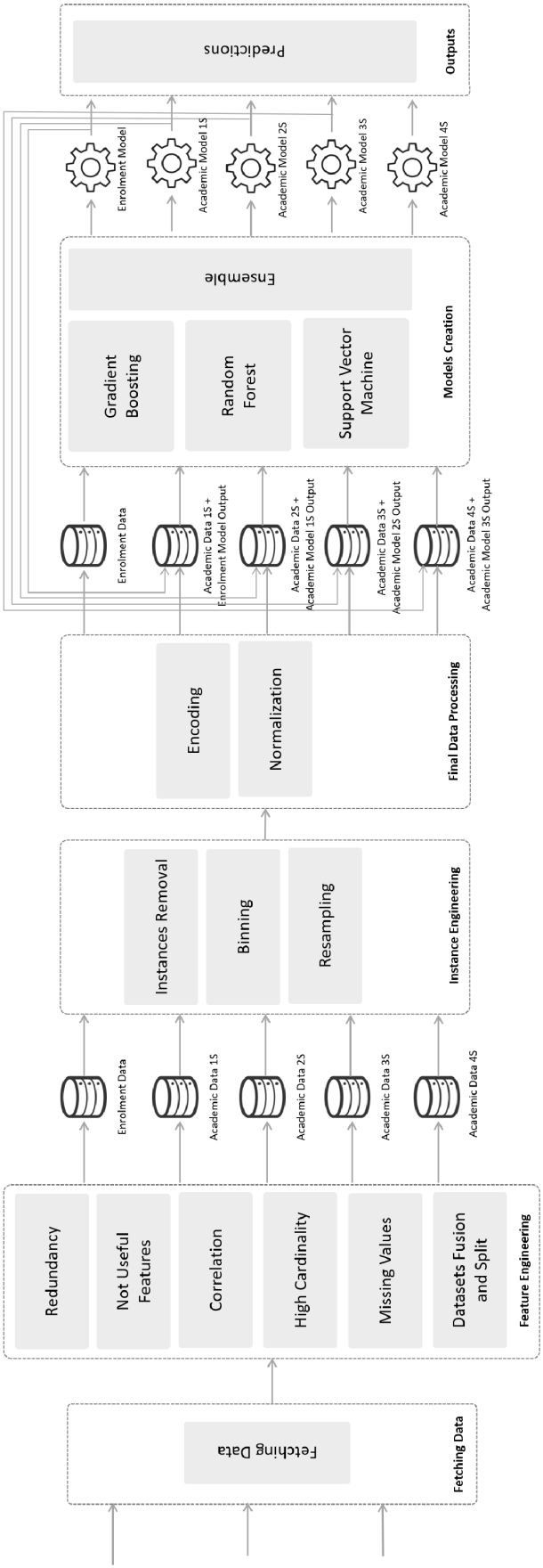


FIGURE 1. Set of processes followed to create the decision support system for engineering students.

main steps, four of them related to *Data Engineering* and two of them related to the creation of models and their evaluation. It is also worth mentioning that this chain of steps constituting this data-driven modelling pipeline may be adapted by the possible elimination, replacement, or adjustment of the steps in accordance with the specific data and problem specifications encountered.

At the end of the process, 5 models are created. The output of these models is the prediction of whether or not a student will complete their studies along with the confidence of that prediction. The main steps of the approach are as follows:

1) **Fetching Data**. Consists of creating a process that connects to the data sources and fetches the data for further processing. In our approach, we extract the data and transform it to a proper format so we can address the specific needs of machine learning workstreams. In-depth details of this part can be found in Subsection IV-A.
2) **Feature Engineering**. Processing data and applying the proper *Feature Engineering* techniques are often overlooked in research papers. Still, if used properly, they can enhance the models even more than other frequently used factors which determine a model's strength. In this part, numerous processes on data, such as dealing with redundancy, significance of features significance, correlation, cardinality features, missing values, and the creation or elimination of features and data fusion, are discussed. In-depth details of this part can be found in Subsection IV-B.
3) **Instance Engineering**. As with *Feature Engineering*, processes related to *Instance Engineering* are not sufficiently taken into account, and their capacity to improve models' performance through the selection and proper manipulation of instances are ignored. Processes such as the removal of unuseful instances, binning, and resampling, are discussed. In-depth details of this part can be found in Subsection IV-C.
4) **Final Data Processing**. Throughout these processes, methods such as Encoding and Normalization are applied to the datasets to ensure that the feeding of data into the algorithms to train models is done in the best possible way. In-depth details of this part can be found in Subsection IV-D.
5) **Model Creation**. This part consists of actually creating the models. Several models are built using different well-known machine learning algorithms, and finally, an ensemble [12] of these independent and diverse models is built with the aim of reducing the generalization error [13]. Additionally, in relation to the different stages of prediction, the output of a predictive model is used as an input for the subsequent model in order to convey the knowledge generated at previous stages. In-depth details of this part can be found in Subsection V-B.
6) **Experimental Results**. Although this part is not directly referred to in Figure 1, before the selection of the models and to consider relevant the outputs produced by the predictive systems, a robust assessment of the model built is performed according to the metrics that are considered suitable for the evaluation of a dropout predictive system such as this. In-depth details of this part can be found in Subsection V-C.

The following two sections detail our methodological approach dealing with the specific challenges addressed, techniques used, and methods applied throughout each of the above steps.

IV. DATA ENGINEERING

A. FETCHING DATA

1) DATA SOURCES

Regarding the data, we ruled out a) the use of private data to avoid problems derived from different privacy policies and regulations; and b) the use of data obtained from LMS to avoid connections with external applications or integration with other tools, i.e. we are limited to the use of academic data only. In terms of the timing of the predictive systems, it is in our interest to create models for several stages including one as early as possible, prior to starting university, and when neither academic nor LMS data is available because teaching has not yet started.

As part of the academic data, in our case, we found 3 different data sources from an engineering school belonging to a Spanish public university:

(a) **Access Records**. This is a data source in the form of a XLSX file, `access.xlsx`, containing the access data from a total of 5 426 observations of 22 features. It coincides with the number of students enrolled in a degree. Note that a student can study for more than one degree, which is very common for undergraduate and graduate programs, and a student can even be enrolled in more than one degree at the same time. These data are known about 2 or 3 months before starting university. However, in some cases, they are not known until just before the beginning of the academic year if students enroll in degrees where vacancies appear at the last minute. After fetching this data source, we name it as the *Enrolment* dataset.
(b) **Grades**. This is a data source, also in the form of a XLSX file, `grades.xlsx`, containing the performance from a total of 194 569 observations of 15 features covering the subjects that comprise 7 engineering degrees and 3 master degrees at a public Spanish University from 2012 to 2019. After fetching this data source, we name it as the *Qualifications* dataset.
(c) **Scholarship**. This is a data source, again in the form of a XLSX file, `scholarship.xlsx`, containing a total of 15 760 observations of 5 features, one of them indicating whether a student has received any type of government scholarship in each of the years they have been enrolled at the university. It is needs

TABLE 2. Set of features describing the access, grades, and scholarship datasets.

Dataset	Feature	Description	Class Description	# Class
Enrolment	**Id**	Student ID	Numerical Values	-
	Degree ID	Unique Degree Identifier	Numerical Values	-
	Degree Name	Degree Name	Categorical: Computer Science Engineering, Civil Engineering, Telecommunication Engineering...	16
	Enrolment Year	First Academic Year of Studies	Categorical: 2007-08, 2008-09...	14
	Closed	Indicate whether the record is closed	Binary: Y, N	2
	Transferred	Indicate whether the record is transferred to other institution	Binary: Y, N	2
	TransferType	Indicates the reason for the transfer of a record	Categorical: Internal, External, Simulataneous	3
	Blocked	Indicate whether the record is blocked	Binary: Y, N	2
	Call	Call for Access	Categorical: June, September...	8
	Call Year	Year of the Call for Access	Categorical: 2007-08, 2008-09, ...	14
	Access ID	Access Type Unique Identifier	Numerical Value	-
	Access Description	Access Type Description	Categorical: University Entrance Exam, Transferred, 25 years of age or older, validated diploma...	11
	SubAccess ID	Subaccess Type Unique Identifier	Numerical Value	-
	SubAccess Description	Subaccess Type Description	Categorical: LOE, LOGSE, LOMCE...	7
	Marks	University Entrance Exam	Numerical Value between 0 and 14	-
	Origin Educational Institution	Origin Educational Institution	Categorical: List of names of the origin High Schools	137
	Birth	Date of Birth	Date	-
	Province ID	Province ID	Numerical Value	-
	Province Name	Province Name	Categorical: Cáceres, Badajoz, Madrid, Toledo...	50
	Municipality ID	Municipality ID	Numerical Value	-
	Municipality Name	Municipality Name	Categorical: Cáceres, Badajoz, Plasencia, Mérida...	434
	Dropout	Indicates if the student has dropped out	Binary: Y, N	2
Qualifications	**Id**	Student ID	Numerical Values	-
	Degree ID	Unique Degree Identifier	Numerical Values	-
	Degree Name	Categorical: Degree Name	Computer Science Engineering, Civil Engineering, Telecommunication Engineering...	16
	Subject ID	Unique Subject Identifier	Numerical Values	-
	Subject Name	Name of the Subject	Categorical: Physics, Linear Algebra, Data Structures and Algorithms, Calculus, Economics and Business...	332
	Year	Year	Numerical Values between 1 and 4	-
	Semester	Semester of the Academic Year	Categorical: 1S, 2S, Annual.	3
	Subject Type ID	Subject Type Unique Identifier	Categorical: B, T, O, P, C, E.	6
	Subject Type Description	Subject Type Description	Categorical:Core, Compulsory, Optional, Internship...	6
	Academic Year	Academic Year	Categorical: 2007-08, 2008-09...	14
	Call	Call of Subject Examination	Categorical: June, September, February...	8
	Mark	Mark	Categorical: Not Taken, Fail, Compensation, Sufficient, Very Good, Outstanding, With Honours	7
	Numerical Mark	Mark in Numerical Format	Numerical Values from 0 to 10	-
	Attempt	Attempt Number	Numerical Values from 1 to 6	-
Scholarship	**Id**	Student ID	Numerical Values	1718
	Academic Year	Academic Year	Categorical: 2007-08, 2008-09...	14
	Degree ID	Unique Degree Identifier	Numerical Values	16
	Degree Name	Degree Name	Categorical: Computer Science Engineering, Civil Engineering, Telecommunication Engineering...	16
	Scholarship	Indicates if the student has a scholarship	Binary: Y, N	2

to be pointed out that, in Spain, in order to obtain a government scholarship for university studies, during the first year it is given automatically and from the second year onwards it is a requirement to pass 65% of the credits enrolled in from the previous year (in the case of engineering, which is the one we are dealing with). In addition, the student must have income limits that are established according to certain family situations. Although it is not possible to speak about strictly private data, it is known that a student with a scholarship cannot belong to a high-income family, having to be below a certain income threshold. After fetching this data source, we name it as the *Scholarship* dataset.

The set of features (description, type, and class) describing each feature in every dataset is listed in Table 2.

2) FETCHING PROCESS

In this phase, the .xls files are transformed into .csv files separated by the character "|". The choice of this character is due to the fact that the character "," can cause some problems in texts or numbers expressed in Spanish format.

This transformation process is carried out through a Python script using the Pandas library. Specifically, all the .xls sheets are read since the data are distributed into multiples of them, and all the sheets belonging to a specific .xls are merged in the same "dataframe". Once the data are in the "dataframe",

they are subsequently exported to .csv format files as previously indicated.

The resulting files are used throughout the project to load data, mainly due to their smaller size and the ease of reading all of their records.

B. FEATURE ENGINEERING

Feature representation has a great impact on improving prediction models [14]. The application of proper *Feature Engineering* techniques, such as deleting, transforming, merging or splitting features, studying feature correlation, or feature cardinality, among others, are fundamental in machine learning [15]. This subsection covers all the aspects related to the application of data preprocessing, data cleaning, and *Feature Engineering* techniques in general, detailing the reasons that have led to its application and how it is carried out. Every process discussed in this subsection aims to transform the original datasets with raw data into the best possible datasets to serve as an input of algorithms and create accurate and reliable predictive models, upon which a decision support system can be relied.

1) REDUNDANT FEATURES

Looking at Table 2 it is clearly visible that certain features are seamlessly integrated with others in a 1 to 1 ratio. This occurs in the features `Degree ID` and `Degree Name`, `Access ID` and `Access Description`, `SubAccess ID` and `SubAccess Description`, `Province ID` and `Province Name`, `Municipality ID` and `Municipality Name` from the *Enrolment* dataset; the features `Degree ID` and `Degree Name`, `Subject ID` and `Subject Name`, and `SubjectType ID` and `SubjectType Name` from the *Qualifications* dataset and; the features `Degree ID` and `Degree Name` from the *Scholarship* dataset.

In each of these cases, one feature must be deleted as both of them provide exactly the same information, so they become redundant. There is an important difference in the features datatype. The 'ID' features are numerical, and the 'Name' or 'Description' features are categorical. In this first stage, the decision is taken to delete the numerical features because there is no ordinal relationship between their classes.

Also in the *Enrolment* dataset, the values of the `Enrolment Year` and `Call Year` features match for more than 99% of the instances. This is because usually the same year that students take their University Entrance Exams, they enroll in college. Thus, having both features would be superfluous. For this reason, we delete the `Call Year` feature to keep the `Enrolment Year`, indicating the academic year in which students start college.

2) UNUSEFUL FEATURES

The *Enrolment* dataset contains the `dropout` feature, which turns out to be the label feature. Additionally, there are other features such as `Closed`, `Transferred`, `TransferType`, and `Blocked` in the same dataset. In fact,

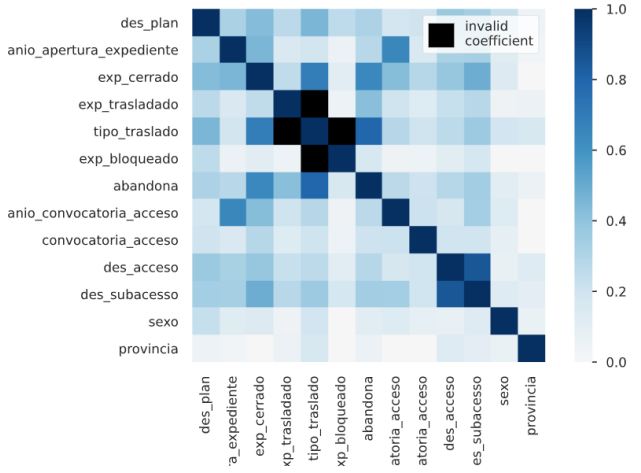


FIGURE 2. Cramér's V correlation matrix.

the `dropout` feature value is calculated from the features enumerated, and hence they can be deleted. This implies the assumption that when students are transferred to another higher education institution, they still count as a dropout for us. Regardless of whether or not the students complete their studies at another institution, we do not have access to that information.

As a result, it makes no sense to keep the `Closed`, `Transferred`, `TransferType`, and `Blocked` features because they are not available at the time the prediction needs to be made.

3) CORRELATION

There are several methods to explain how one or more variables are related to each other and by calculating their correlation it is possible to examine their degree of relationship. Figure 2 shows a correlation matrix calculated using the Cramér's V method [16] which measures the association between two discrete features assigning a value between 0 and 1. It is based on the Chi-squared statistic method [17].

Looking at Figure 2) there is a high degree of correlation between the `Access Description` and `Subaccess Description` features from the *Enrolment* dataset. This correlation exists because a) there is a hierarchical relationship between the features where `Subaccess Description` is the lowest level in the group hierarchy; and b) when the `Access Description` feature has a class that does not have more than one possible associated value in the `Subaccess Description` feature, the values coincide. All of this leads to the removal of the `Subaccess Description`, the lowest in the hierarchy.

4) HIGH CARDINALITY FEATURES

The `Origin Educational Institution` feature from the *Enrolment* dataset exhibits High Cardinality which is liable to hinder the ability of inferring knowledge from the machine learning algorithms. In Figure 3 it can be seen that there are 137 distinct values (and a large number of missing

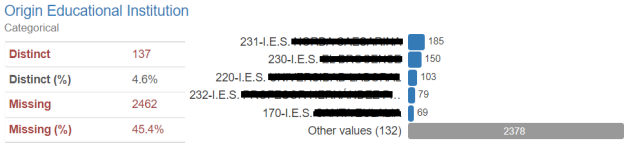


FIGURE 3. `Origin educational institution feature exploration`.

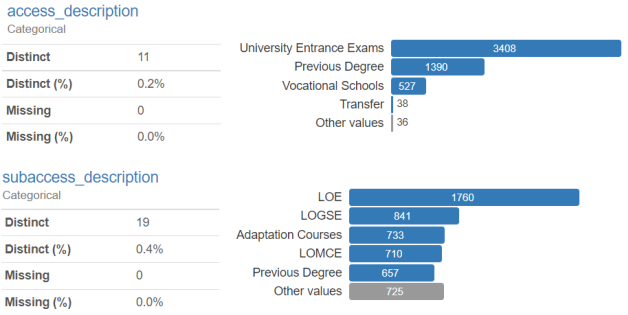


FIGURE 4. `Access description` **and** `Subaccess description` **cardinality.**

values, which are discussed later). The extensive fragmentation of this feature's classes can lead to high-dimensional spaces or sparsity of data [18]. To avoid problems such as these, which may negatively affect the model's accuracy, the `Origin Educational Institution` feature is removed.

Nevertheless, it is not a decision taken lightly to remove this feature, given that the power of predictability for this feature is potentially high. However, on top of high cardinality, there is a large number of missing values 45.4%, and the classes of the feature are not homogeneous, as students can originate from different types of institutions such as high schools, universities or vocational training schools, among others. All of the above reaffirms the decision to remove the feature.

Also concerning the features cardinality, the `Subaccess Description` was removed. In fact, we removed it from the previous subsection due to correlation issues, but before doing so, we considered grouping both hierarchized features into just one. We discarded this option not only due to the even higher cardinality of the resultant feature, but also because there are predominant classes in both features, which could lead to a highly imbalanced distribution of classes. In addition, the feature may not be representative and problems such as the curse of dimensionality may arise. The exploration of these features can be seen in Figure 4.

The `Municipality` feature also exhibits High Cardinality problems. As happens with the `Origin Educational Institution` feature, it is supposed to be significant enough to compensate for the increase in dimensions involved in deciding to leave it in the dataset. Accordingly, this feature is kept.

5) MISSING VALUES

Missing values are common in real world datasets, and ours is no exception. Figure 5 shows the missing values of the

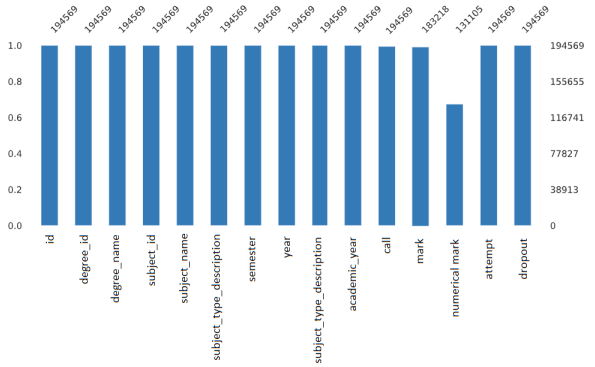


FIGURE 5. **Missing values in the *Qualifications* dataset.**

Qualifications dataset. The *Scholarship* and *Enrolment* datasets have no missing values with the exception of the `Origin Educational Institution` and `Transfer type` features, both of which have already been removed from the datasets due to other reasons previously discussed. The *Qualifications* dataset has missing values in the `Numerical Mark` feature. With a closer glance one can see that the missing values correspond to the instances in which the value of the `Mark` feature is "Not Shown", or similarly, the student has not taken the exam.

Nothing is done to solve the missing values issue at this stage, even though we are fully aware of the situation, because there are no problems associated with this circumstance. This is due to the lower level of granularity of the *Qualifications* dataset as is described further on in the next subsection (Subsection IV-B6).

6) DATASETS FUSION

Until now, we have worked with the three datasets in parallel, applying different data processing and *Feature Engineering* techniques. At this point it is essential to merge the datasets to create 5 predictive models. To reiterate, we aim to create 5 models at 5 different stages: at enrolment and at the end of each of the first four semesters.

For the creation of the first model, which corresponds to the time of enrolment, there is data available from the *Enrolment* and *Scholarship* datasets so they must be merged. Two out of five features regarding the *Scholarship* dataset are in the *Enrolment* dataset so it is easy to join the datasets together using these features (`Id`, `Degree Name`). Note that in the *Scholarship* dataset, a student can appear more than once because there is one record per academic year. For the first model there is only information available about the enrolment year.

For the creation of the subsequent models, the three datasets must be merged. We already have merged the *Enrolment* and *Scholarship* datasets, so now we must merge the *Qualifications* dataset as well. It is the case that each observation in *Enrolment* dataset is at a granularity level of "student on a degree" and each observation in the

Qualifications dataset has the granularity level of ''exam for a subject taken''. As such, the data from the *Qualifications* dataset must be grouped in a certain way in order to be merged with the data from the *Enrolment* and *Scholarships* datasets. Some new features are created from the *Qualifications* dataset at a ''student on a degree'' granularity level. These new features, which are ratios, can then be merged with the *Enrolment* dataset as they now have the same granularity level. These new calculated features are:

- `Success Rate`. This metric measures student performance up to the first, second, third, or fourth semester, depending on the model to be trained. It is calculated by dividing the number of subjects the student has passed by the total number of subjects the student has enrolled in. It is defined by the following equation:

$$Success\ Rate = \frac{Subjects\ passed}{Subjects\ Enroled} \tag{1}$$

- `Not Shown Rate`. This metric measures the percentage of subjects a student has not taken any exam in up to the first, second, third, or fourth semester, depending on the model to be trained. It is calculated by dividing the number of subjects in which the student has not taken any exam by the total number of subjects the student has enrolled in. It is defined by the following equation:

$$Not\ Shown\ Rate = \frac{Subjects\ not\ taken}{Subjects\ Enroled} \tag{2}$$

- `Median Marks`. Median of the marks obtained by a student up to the first, second, third, or fourth semester, depending on the model to be trained. We use the median and not the mean as a measure of dispersion because the latter can be easily affected by *outliers*, so it would not represent the grades that the person usually gets throughout their degree.

The following features from the *Qualification* dataset are used to calculate these metrics: `Semester`, `Academic Year`, `Numerical Mark`. Other features that exist in the *Qualification* dataset such as `Year`, `Mark`, `Attempt`, `Subject Name` or `Call` are not used to calculate metrics and they have been discarded even though the calculation of other metrics such as *Average Mark* or *Average Attempts*, among others, could be of greater interest.

C. INSTANCE ENGINEERING

This subsection discusses the application of *Instance Engineering* techniques in detail. The careful selection of instances is crucial for improving the predictive models' performance [19]. Not every observation makes a positive contribution to the dataset and hence it is important to detect and remove them to train the models with the instances that positively contribute to the model learning capacity.

1) INSTANCE REMOVAL

There are some instances that may have a detrimental impact on model learning capacity. In our datasets, there are several instances that, for a variety of reasons, find themselves in this situation.

The three datasets contain instances corresponding to old degree programs that are no longer being implemented. Since we want to create a decision support system with a predictive model to prevent dropout in current degrees, instances from old degrees (unless they have a current equivalent degree) are not valid as they may add noise. These instances are removed.

In the *Enrolment* dataset, some instances correspond to people for whom we do not have complete information. Furthermore, there are students for whom we do not have information from their first year until the year they finish or dropout of college. Finally, students who started the degrees in the academic years '2017-18', '2018-19', '2019-20', '2020-21' have not yet been able to complete their studies and there is no certainty whether they will finish them or not. For these reasons, we have removed the instances of those students whose number of core subjects taken is less than 10 from the datasets, in order to identify those students for whom we lack information from their first years, as well as removing the instances of those students who enrolled less than 5 years ago.

2) BINNING

Binning is commonly used in machine learning to transform a numerical feature with continuous values into a categorical feature comprising continuous values into determined ranges. This technique has been applied to the `birth` feature from the *Enrolment* dataset. After many experiments, we decided to obtain the year of birth of the students by dividing it into ranges of 5 years to obtain a non-numerical feature. To do this, a customized lambda in python has been developed that extracts the year through the DateTime data type and divides it into ranges using the *cut* function of the *pandas* library. [20].

3) RESAMPLING

Commonly, classification problems dealing with real-world datasets may suffer from class imbalance problems. Class imbalance occurs when the number of instances of each class from the label feature is highly unbalanced, there being many occurrences of certain classes and few others. The severity of the problem will vary depending on how extreme the imbalance between the classes is. Many techniques can be used to solve the problem and allow classes to be equally represented in the dataset. There are simple techniques that consist of randomly duplicating instances from the minority class or randomly deleting instances from the majority class until we get a ratio where classes are equally represented. In contrast, there are more complex techniques that consist of synthesising new instances from real instances.

In our dataset, from a total of 1 418 observations (after applying the *Feature Engineering* and *Instance Engineering* techniques introduced above), there are 783 instances corresponding to students who drop out of college and 635 to students who complete their studies. To equally distribute the classes with a 1:1 ratio, we use a combination of SMOTE and

TABLE 3. Final features of the datasets.

				Datasets				
Feature	**Source**	**Data Type**	**Observations**	**Enrolment**	**1S**	**2S**	**3S**	**4S**
Id	Enrolment	Numerical	For student identification only (Not training)	✓	✓	✓	✓	✓
Degree Name	Enrolment	Categorical	7 Classes	✓	✓	✓	✓	✓
Enrolment Year	Enrolment	Categorical	9 Classes	✓	✓	✓	✓	✓
Access Description	Enrolment	Categorical	2 Classes	✓	✓	✓	✓	✓
Call	Enrolment	Categorical	2 Classes: Ordinary and Extraordinary	✓	✓	✓	✓	✓
Mark (Entrance Exam)	Enrolment	Numerical	Min:5, Max: 13.464, Mean: 7.36. Std: 1.736	✓	✓	✓	✓	✓
Birth	Enrolment	Categorical	Binning transformation in 5 year ranges	✓	✓	✓	✓	✓
Municipality	Enrolment	Categorical	229 Classes (High Cardinality)	✓	✓	✓	✓	✓
Dropout	Enrolment	Categorical	Target Value to Predict (Label)	✓	✓	✓	✓	✓
Scholarship	Scholarship	Numerical	Discrete. 1 (Yes), 0 (No)	✓	✓	✓	✓	✓
Success Rate	Qualifications	Numerical	Calculated Value		✓	✓	✓	✓
Not Shown Rate	Qualifications	Numerical	Calculated Value		✓	✓	✓	✓
Marks Median	Qualifications	Numerical	Calculated Value		✓	✓	✓	✓
Enrolment Model Output	Enrolment Model	Numerical	Discrete. 1 (Dropout), 0 (No dropout)		✓			
1S Model Output	1S Model	Numerical	Discrete. 1 (Dropout), 0 (No dropout)			✓		
2S Model Output	2S Model	Numerical	Discrete. 1 (Dropout), 0 (No dropout)				✓	
3S Model Output	3S Model	Numerical	Discrete. 1 (Dropout), 0 (No dropout)					✓

Tomek Links methods [21], [22]. It takes observations from the datasets and looks for other observations that are closer (neighbours). Once a number of neighbours are found, new observations are synthetically created spatially in the conjunction of the picked neighbours. New instances belonging to the majority class are discarded to making a less noisy decision boundary. At the end of the process, 148 synthetic instances from the minority class are generated.

D. FINAL PROCESSING

After applying all the *Feature Engineering* and *Instance Engineering* techniques discussed in the previous subsections, the datasets as read from the data sources have changed significantly. Table 3 shows the features present in each of the datasets that are going to be used to build predictive models: prior to enrolment, at the end of the first semester, at the end of the first year, at the end of the third semester, and at the end of the second year. Each row of the table corresponds to a feature and provides information such as the source of the feature, its data type and some observations inherent to its form.

The final columns of Table 3 indicate whether the feature is present in each of the 5 datasets. The *Enrolment Dataset*, which is used to build the prior enrolment predictive model, only has data related to the students and scholarships. The first semester dataset (*1S dataset*) also incorporates features related to the students' performance in the first semester and the output of the previous model. The first year dataset *2S dataset* updates the student data, calculating performance metrics regarding the first 2 semesters (during the first year) and also takes the output of the previous model as an input. Finally, The third semester dataset (3S Dataset) updates the student data calculating performance metrics regarding the first 3 semesters and also takes as input the output of the previous model (the 2S model). The same happens with the 4th semester dataset.

However, even after the application of all of these techniques prior to building the models, the datasets need additional processing to deal with aspects such as *Encoding* and *Normalization*.

1) ENCODING

Many algorithms are unable to process categorical values. This requires the application of processing techniques for them to work with non-numerical features. However, the transformation from categorical features to numerical features is not exempt from risk and biases that can affect the models' performance. There are many encoding techniques that can be employed, the best known being *Label Encoding* and *One-Hot Encoding*. The former assigns an integer value to each of the feature values, while the latter consists of creating n new binary features [10], for each given feature with n classes.

The features that need to be encoded in our datasets are: `Degree Name`, `Enrolment Year`, `Access Description`, `Call`, `Birth`, and `Municipality`. The `dropout` feature is the target so it does not need to be encoded. We decide to apply the *One-Hot Encoding* because there is no particular numerical order relationship between the classes of the features. Each of these features will therefore be deleted, creating new features which correspond with each of their classes. In each instance, the class corresponding with the new features created will be marked 1 and the rest 0.

2) NORMALIZATION

Data normalization is a common practice in machine learning that consists of transforming numerical columns to a common scale. This is done to avoid features with high values dominating the learning process, even though they are no

more important than others when determining the output of a model.

There are many possible scalers that can be applied. We make use of a simple scaler called *MinMax Scaler*. This scales each feature individually to a given range and we set a [0, 1] range. When using this scaler, the shape of the distribution remains unchanged. If there happened to be *outliers*, they would not be affected. This is not usually ideal, but, in our case, there are no significant *outliers* in any of the features, and as such, it is a proper scaler to be applied.

3) TRAINING AND TEST DATASETS

This process consists of splitting the datasets into 2 parts: the training dataset containing approximately 75% of the instances of the total dataset, and the test dataset making up the remaining 25% of the instances.

This module is carried out at this point, and not before, for one reason: to ensure that the experiment's results are as objective as possible. If each algorithm divides the dataset by itself randomly, the random seeds would be different, and this would have potentially negative implications for any objective comparison of the models created by each of those modules.

After all these processes, the datasets are primed to serve as inputs for the learning algorithms.

V. MODEL CREATION AND EXPERIMENTATION

This section describes the machine learning algorithms used to build the predictive systems. After that, the predictive models are evaluated according to certain metrics that are considered relevant for measuring the proper functioning of the models in line with the addressed problem: identify as many students at risk of dropout as possible.

A. MODEL CREATION

The number of classification algorithms available is large and hence the possible values to assign to the algorithm hyperparameters enormous. All of these considerations clearly affect the performance of the models.

In our work, classification algorithms available on *sklearn* [23], and which have been proved successful in related work, have been selected. In addition to *sklearn* we take advantage of the *pandas* [24] library to handle data as when performing *Data Engineering* operations.

The algorithms employed are: *Gradient Boosting*, *Random Forest*, and *Support Vector Machine*, which can be defined as follow:

- *Gradient Boosting*. This consists of constructing many *Decision Tree* models to create strong learners. *Gradient Boosting* uses *Loss Function* as a measure of indicating how good the model's coefficient is. The idea is to use a hypothesis for weak learners and refocus on the examples that learners find difficult to classify [25].
- *Random Forest*. It combines many *Decision Tree* algorithms built using the same data after the application of random changes, but before the creation of each independent tree. The most popular of their outputs is then taken. The generalization error for forests converges to a limit as the number of trees in the forest becomes large [26].
- *Support Vector Machine*. This algorithm classifies classes using a separator. It takes data to high multidimensional spaces where it is possible to find a hyperplane that separates not otherwise linearly separable classes. From all the hyperplanes possible, the one that offers a greater margin between classes is selected [27].

In the search for the best results, different tests are carried out on the configuration of the hyperparameters that configure the performance and determine the predictive capacity that the algorithms can achieve [28]. This process, known as *hyperparameter tuning* is conducted using the *GridSearchCV* function contained in the *model_selection* package from *Scikit-Learn*. This tries out several combinations of hyperparameters and evaluates the resultant model using *Cross-Validation* [29].

In addition to creating individual models, the creation of an ensemble of the models built is also considered, since its ability to improve the robustness and accuracy of decision systems [12], [30] is already known. The proposed *Ensemble* takes the *Gradient Boosting*, *Random Forest*, and *Support Vector Machine* models' outputs and calculates its own output by proportionally distributing the weight of each model.

With regard to the dataset used to train the models at each stage there are two main differences as may be observed in Table 3:

- The enrolment model output is used as a feature of the first-semester model; the first-semester model output is used as a feature of the second-semester model; and so on.
- Starting with the first-semester model, the `Success Rate`, `Not Shown Rate`, and `Marks Median` features are incorporated as features in the dataset and then fed into the algorithms. These metrics are calculated independently for each one. In the first-semester model, they are calculated with the marks obtained in the first semester; in the second-semester model, they are calculated with the marks obtained in the first year; and so on.

B. EVALUATION METRICS

Many evaluation metrics can be considered when determining the suitability of a model to be applied to a particular problem. In this work, we use the following metrics to validate our models and to determine their validity.

- **Accuracy**. This is a basic metric that indicates the correctly predicted instances. It can be defined as:

$$Accuracy = \frac{Number\ of\ correctly\ predicted\ observations}{Total\ number\ of\ observations} \quad (3)$$

- **Confusion Matrix**. This is a table layout where each row represents the number of instances of each class and

TABLE 4. Confusion matrix.

		Enrolment Model		Semester 1 Model		Semester 2 Model		Semester 3 Model		Semester 4 Model	
Gradient Boosting	Yes	102	62	134	25	126	33	130	29	134	20
	No	39	115	41	117	28	130	23	124	15	135
Random Forest	Yes	104	60	129	30	125	34	138	21	136	18
	No	40	114	37	121	31	127	18	129	17	133
Support Vector Machine	Yes	108	56	119	40	125	34	134	22	130	16
	No	44	110	32	126	27	131	18	133	12	149
Ensemble	Yes	89	75	125	34	125	34	132	27	133	21
	No	30	124	27	131	27	131	20	127	13	137
		Yes	No	Yes	No	Yes	No	Yes	No	Yes	No

TABLE 5. Evaluation results.

			Precision		Recall	
		Accuracy	**Dropout**	**Not Dropout**	**Dropout**	**Not Dropout**
Enrolment Model	Gradient Boosting	68.239	62.195	74.675	**72.340**	64.972
	Random Forest	68.553	63.415	74.026	72.222	65.517
	Support Vector Machine	68.553	65.854	71.429	71.053	66.265
	Ensemble	66.981	54.268	80.519	74.790	62.312
1S Model	Gradient Boosting	79.180	84.277	74.051	76.571	82.394
	Random Forest	78.864	81.132	76.582	77.711	80.132
	Support Vector Machine	77.287	74.843	79.747	78.808	75.904
	Ensemble	80.757	78.616	82.911	**82.237**	79.394
2S Model	Gradient Boosting	80.757	79.245	82.278	81.818	79.755
	Random Forest	79.495	78.616	80.380	80.128	78.882
	Support Vector Machine	80.757	78.616	82.911	82.237	79.394
	Ensemble	80.757	78.616	82.911	**82.237**	79.394
3S Model	Gradient Boosting	83.007	81.761	84.354	84.967	81.046
	Random Forest	87.255	86.792	87.755	**88.462**	86.000
	Support Vector Machine	86.971	85.897	88.079	88.158	85.806
	Ensemble	84.641	83.019	86.395	86.842	82.468
4S Model	Gradient Boosting	88.487	87.013	90.000	89.933	87.097
	Random Forest	88.487	88.312	88.667	88.889	88.079
	Support Vector Machine	90.879	89.041	92.547	**91.549**	90.303
	Ensemble	88.816	86.364	91.333	91.096	86.709

each column represents the class that has been predicted by the model [31]. A confusion matrix is created for each algorithm selected at every stage of its application and allows the analysis of the models' performance beyond *accuracy*, looking at the details of the predictive labels according to the actual labels and performs a deeper analysis of a model's behaviour.

- **Precision**. This indicates the proportion of predicted positives. For our case, the *precision* metric determines a given number of students who drop out and how many who do not drop out have been misclassified. Using the confusion matrix, it can be calculated as:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

where

  - TP = true positives,
  - FP = false positives.

- **Recall (Sensitivity)**. This indicates the proportion of positives predicted as positives. For our case, the *recall* metric determines how many dropouts are identified out of the total number of dropouts. Or put another way, the number of students who drop out and are not identified. Using the confusion matrix, it can be calculated as:

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

where

  - TP = true positives,
  - FN = false negatives.

C. EXPERIMENTAL RESULTS

This subsection shows the results of the experiments carried out with the aim of selecting the best performing models for our purpose, *i.e.*, thus identifying the students at risk of dropout. Table 4 and Table 5 show the relevant metrics used to determine the validity of the results and the most suitable models. The former shows the *Confusion Matrix* and the latter the *Accuracy*, *Precision*, and *Recall* evaluation metrics.

Of particular interest is the predictive ability regarding the risk of dropping out before starting college. The earlier a student's risk of dropping out is known, the earlier action can be taken. In this respect, analysing the *enrolment models* metrics, we can see that these models' accuracy reaches values close to 70%, which is a good result bearing in mind that there is little information available concerning the student and additionally, personal data have not been used to train the models.

In any case, the value that we identify as most important and the one that is the basis for decision-making when identifying the most suitable models is that of recall because it identifies the maximum number of students dropping out of the total number of students who dropout. Thus, looking at the *recall* column in Table 5 we can see that the best performing model at the time of enrolment is the *ensemble*, which is capable of identifying almost 75% of students who will drop out.

It is true that if the *Ensemble* model reaches a *recall* of 74.79%, other metrics such as *precision* suffer, grouping several students who do not dropout with those who dropout. This results in a high cost in terms of applying dropout prevention policies as well as inconvenience for a considerable number of students who may be mistakenly classified as showing dropout risk when they that is not the case. Accordingly, the *Gradient Boosting* model should be selected, being the most appropriate model for this phase, since although it is somewhat less effective in identifying dropouts in comparison with the *Ensemble* model (72.340 vs. 74.79), it is able to include fewer non-dropouts among the students at risk, showing a higher precision of 62.195 instead of 54.268 for the *Ensemble* model. As a result, the *Gradient Boosting* model is the ideal model to be used at the time of enrolment.

For the first and second-semester models, where the students' initial marks are already available, the models significantly improve their results with the *Ensemble* model performing best at this stage. The first-semester model increases the *accuracy* of the best model by 10 percentage points for *recall* and by 12 percentage points for the time of enrolment. In addition, *precision* rises by 16%. The model selected as appropriate at this stage is the *Ensemble* model. The second-semester model maintains similar metrics to the previous model even when the results from the *Gradient Boosting* and *Random Forest* model improve.

The third and fourth-semester models continue an upward trend in terms of the values obtained by the performance metrics, as is logical. The fourth-semester *recall* is 91.549 which is an excellent value although it is true that at this stage when students have already spent two years at college, it is fairly easy to anticipate whether they are going to finish the degree or dropping out. An even higher value of the *precision* metric at this stage allows clear identification of the students at risk of dropout without having many false positives and the problems that these entail.

Note that there is no predominant algorithm in any of the stages. Furthermore, *accuracy* is stable in the sense that there is no predominant algorithm, and the values of these metrics do not even differ by 2 points between the best and the worst model in each phase with the exception of the third-semester model. The *recall* metric has more obvious fluctuations of up to 4 points, and this is the main metric for our purposes. However, there is still no algorithm that consistently obtains better results. It may seem that the nature of data does not make it ideal to be modeled in a specific way and that the problems and data vary from stage to stage. This can be explained by the number of instances to test the result of the models being affected throughout the successive stages (319 - 317 - 317 - 306 - 304, from the enrolment model to the fourth-semester model) because the instances where students have already dropped out are removed.

VI. CONCLUSION AND FUTURE WORK

In this article, we address the problem of creating predictive systems to anticipate the dropout risk of Higher Education students at different stages of their studies, starting at enrolment time up to the end of the fourth semester in semester intervals. By identifying the students with a high degree of probability of dropping out, it is possible to design and apply dropout prevention policies effectively. This study focuses on students from engineering degrees using real-life data.

The whole process from fetching data to the construction and evaluation of the predictive models has been depicted. This involves the creation of guidelines that are based on data that all the Higher Education Institutions have about their students. Although they may be adapted to each institution, their high degree of homogeneity allows the work carried out to be easily reproduced. Indeed, the processes are organised and detailed to facilitate the understanding of the proposal and encourage its implementation.

The main parts of the proposal include 1) How to fetch data from data sources and transform them into meaningful datasets; 2) The application of *Feature Engineering* techniques that transform data which is better represented, thus having a great impact on the learning of predictive systems; 3) The application of *Instance Engineering* for the careful selection of instances that makes a positive contribution to the models' performance. 4) Delve into data engineering processes such as encoding or normalization to train the models in the best possible way; 5) Design and develop models using well-known algorithms such as *Gradient Boosting*, *Random Forest* and, *Support Vector Machine* leading to the creation of an *Ensemble* from their outputs; and finally 6) Carefully select the appropriate evaluation metrics to properly interpret the experimental results carried out. All of this leads to implementing the best performance model for each of the stages in which we can apply dropout prevention policies, from enrolment time and on a semester basis until the end of a student's second year.

Other important aspects of the work to be highlighted include: a) The implementation of a cascade connection for the models' outputs in such a way that the output of each model serves as an input to the subsequent models contributing to the use of the knowledge generated at each step to improve the subsequent ones; b) the comparison of those algorithms which behave better in the different prediction stages (at enrolment, in the first semester, ...).

With regard to the results obtained, they are promising, given the limited data available to us. Focusing on the time of enrolment, which is the earliest possible time of intervention when the institution has no relation whatsoever with the

students beyond enrolment in a particular degree, the predictive model created can detect more than 72% of the students that will dropout. This provides a huge range of action to start working with the application of prevention policies. By the end of the first semester, it is possible to detect more than 82% of students that will dropout without misclassifying students that are not at dropout risk (Precision = 82.91%). These models, that allow the detection of abandonment at very early stages, are tremendously important. In more advanced stages, even though they do not provide such early prediction, the results improve and detect up to 91.5% of students that will discontinue their studies at the end of the fourth semester.

In future works, it would be of interest to improve the models created and validate their suitability by following these practices:

- Expand the scope of action to study the performance of these guidelines in other areas in higher education such as Education, Humanities, Business, and/or Experimental Sciences, among others.
- Study the impact of the calculation of new metrics at student granularity level in the *Qualifications* dataset.
- Design a follow-up plan to incorporate or modify data engineering techniques based on the results obtained by the application of the dropout prevention policies.

REFERENCES

[1] F. Pérez, J. Aldás, R. Aragón, and I. Zaera, Eds., *Indicadores Sintéticos de las Universidades Espanolas*. San Luis Potosí, Mexico: Fundación BBVA, 2019.

[2] L. P. Macfadyen and S. Dawson, "Mining LMS data to develop an 'early warning system' for educators: A proof of concept," *Comput. Educ.*, vol. 54, no. 2, pp. 588–599, 2010.

[3] F. Marbouti, H. A. Diefes-Dux, and K. Madhavan, "Models for early prediction of at-risk students in a course using standards-based grading," *Comput. Educ.*, vol. 103, pp. 1–15, Dec. 2016.

[4] C. C. Gray and D. Perkins, "Utilizing early engagement and machine learning to predict Student outcomes," *Comput. Educ.*, vol. 131, pp. 22–32, Apr. 2019.

[5] V. L. Miguéis, A. Freitas, P. J. V. Garcia, and A. Silva, "Early segmentation of students according to their academic performance: A predictive modelling approach," *Decis. Support Syst.*, vol. 115, pp. 36–51, Nov. 2018.

[6] Y. Chen, Q. Zheng, S. Ji, F. Tian, H. Zhu, and M. Liu, "Identifying at-risk students based on the phased prediction model," *Knowl. Inf. Syst.*, vol. 62, no. 3, pp. 987–1003, Mar. 2020.

[7] S. Helal, J. Li, L. Liu, E. Ebrahimie, S. Dawson, D. J. Murray, and Q. Long, "Predicting academic performance by considering student heterogeneity," *Knowl.-Based Syst.*, vol. 161, pp. 134–146, Dec. 2018.

[8] A.-S. Hoffait and M. Schyns, "Early detection of University students with potential difficulties," *Decis. Support Syst.*, vol. 101, pp. 1–11, Sep. 2017.

[9] H. A. Mengash, "Using data mining techniques to predict Student performance to support decision making in university admission systems," *IEEE Access*, vol. 8, pp. 55462–55470, 2020.

[10] A. J. Fernandez-Garcia, R. Rodriguez-Echeverria, J. C. Preciado, J. M. C. Manzano, and F. Sanchez-Figueroa, "Creating a recommender system to support higher education students in the subject enrollment decision," *IEEE Access*, vol. 8, pp. 189069–189088, 2020.

[11] A. J. Fernández-García, L. Iribarne, A. Corral, and J. Criado, "A comparison of feature selection methods to optimize predictive models based on decision forest algorithms for academic data analysis," in *Trends and Advances in Information Systems and Technologies*, Á. Rocha, H. Adeli, L. P. Reis, and S. Costanzo, Eds. Cham, Switzerland: Springer, 2018, pp. 338–347.

[12] V. Kotu and B. Deshpande, "Data science process," in *Data Science*, 2nd ed, V. Kotu and B. Deshpande, Eds. San Mateo, CA, USA: Morgan Kaufmann, 2019, pp. 19–37.

[13] S. Theodoridis, "Learning in parametric modeling: Basic concepts and directions," in *Machine Learning*, 2nd ed. S. Theodoridis, Ed. New York, NY, USA: Academic, 2020, pp. 67–120.

[14] K. Ramasubramanian and A. Singh, *Feature Engineering*. Berkeley, CA, USA: Apress, 2017, pp. 181–217.

[15] A. J. Fernández-García, L. Iribarne, A. Corral, J. Criado, and J. Z. Wang, "A recommender system for component-based applications using machine learning techniques," *Knowl.-Based Syst.*, vol. 164, pp. 68–84, Jan. 2019.

[16] H. CraméR, *Mathematical Methods of Statistics (PMS-9)*. Princeton, NJ, USA: Princeton Univ. Press, 1999.

[17] N. Mantel, "Chi-square tests with one degree of freedom; extensions of the Mantel–Haenszel procedure," *J. Amer. Stat. Assoc.*, vol. 58, no. 303, pp. 690–700, 1963.

[18] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning: Data mining, inference, and prediction," *Int. Stat. Rev.*, vol. 77, no. 3, p. 482, 2009.

[19] B. Baesens, S. Höppner, and T. Verdonck, "Data engineering for fraud detection," *Decis. Support Syst.*, vol. 150, Jan. 2021, Art. no. 113492.

[20] *Pandas-Dev/Pandas: Pandas*, The Pandas Development Team, Road Dallas, TX, USA, Feb. 2020.

[21] G. Batista, A. Bazzan, and M.-C. Monard, "Balancing training data for automated annotation of keywords: A case study," in *Proc. 2nd Brazilian Workshop Bioinf.*, Macaé, Brazil, Dec. 2003, pp. 10–18.

[22] T. Sasada, Z. Liu, T. Baba, K. Hatano, and Y. Kimura, "A resampling method for imbalanced datasets considering noise and overlap," *Proc. Comput. Sci.*, vol. 176, pp. 420–429, Jan. 2020.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[24] W. McKinney, "Pandas: A foundational Python library for data analysis and statistics," *Python High Perform. Sci. Comput.*, vol. 14, no. 9, pp. 1–19, 2011.

[25] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[26] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[27] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul. 1998.

[28] M. Feurer and F. Hutter, *Hyperparameter Optimization*. Cham, Switzerland: Springer, 2019, pp. 3–33.

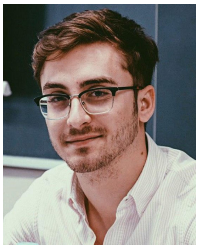
[29] P. Refaeilzadeh, L. Tang, and H. Liu, *Cross-Validation*. Boston, MA, USA: Springer, 2009, pp. 532–538.

[30] C. Zhang and Y. Ma, *Ensemble Machine Learning: Methods and Applications*. New York, NY, USA: Springer-Verlag, 2012.

[31] K. M. Ting, *Confusion Matrix*. Boston, MA, USA: Springer, 2017, p. 260.

ANTONIO JESÚS FERNÁNDEZ-GARCÍA received the Ph.D. degree in computer science from the University of Almería, in 2019. In September 2020, he joined as an Associate Professor with the Universidad Internacional de la Rioja (UNIR). He is currently a Researcher with the Quercus Software Engineering Group, Department of Computer Science, University of Extremadura (UEX), and the Applied Computing Group, University of Almería (UAL). He has published more than 15 scientific publications in journals and international conferences. His research interests include recommender systems, machine learning, artificial intelligence, data mining, data engineering, and software engineering.

JUAN CARLOS PRECIADO received the Ph.D. degree in computer science from the University of Extremadura (UEX), in 2008. He is currently a Professor and a member of the Quercus Software Engineering Group, Department of Computer Science, UEX, where he was also the Vice-Rector for several years. His research interests include model-driven development and web and data engineering, where he has published around 100 papers in the software engineering field.



FRAN MELCHOR received the bachelor's degree in software engineering from the University of Extremadura, where he is currently pursuing the master's degree in data science. He was hired as a Research Support Technician at the University of Extremadura. His research interests include software engineering, artificial intelligence, data science, and information systems.

ROBERTO RODRIGUEZ-ECHEVERRIA received the Ph.D. degree from the University of Extremadura, Spain. He is currently a member of the Quercus Software Engineering Group and a Professor of computer languages and systems at the University of Extremadura. He has published more than 50 scientific publications in journals and international conferences. His research interests include software engineering, web engineering, model-driven engineering, legacy software modernization, and end-user development.

JOSÉ MARÍA CONEJERO received the Ph.D. degree in computer science from the Universidad de Extremadura, in 2010. He is currently an Assistant Professor with the University of Extremadura. He is the author of more than 50 papers of journals and conference proceedings and has also participated in different journals and conferences as a member of the program committee. His research interests include the aspect-oriented software development, requirements engineering, and model-driven development or ambient intelligence.

FERNANDO SÁNCHEZ-FIGUEROA received the Ph.D. degree in computer science from UEX. He is currently a Professor with the Department of Computer Science, UEX. He is the coauthor of more than 100 publications related to software engineering. His research interests include web engineering, big data visualization, and MDD.