Form@re
Open Journal per la formazione in rete

esperienze / riflessioni

Predictive models for effective policy making against university dropout

Modelli predittivi per politiche efficaci contro l'abbandono dell'università

Stefano Pio Zingaro[a], Agnese Del Zozzo[b], Francesca Del Bonifro[c], Maurizio Gabbrielli[d]

[a] *Università di Bologna / INRIA, stefanopio.zingaro@unibo.it*
[b] *Libera Università di Bolzano / Università di Trento, agnese.delzozzo@gmail.com*
[c] *Università di Bologna / INRIA, francesca.delbonifro@unibo.it*
[d] *Università di Bologna / INRIA, maurizio.gabbrielli@unibo.it*

Abstract

The mere development of a software to predict University dropout is not sufficient for its effective implementation in the academic context. In order to exploit it as a tool supporting decision-making, such a software should be provided with information necessary for its integration into the decision-making processes of University governance. In this work, we present a predictive tool, at the state of the art, providing a functional description of its integration through practical examples. In addition, we propose a simplified scheme to guide the reasoning on the software, which we structure according to the following processes: the learning of the machine, the choice of the representation of dropout and the interpretation of the results. Finally, we share some considerations addressing education designers and institutional decision-makers on the management of interventions inspired by the prediction of freshmen academic outcome.

Keywords: dropout; decision support system; predictive model.

Sintesi

Lo sviluppo di un software per predire l'abbandono della carriera universitaria non è sufficiente affinché questo venga implementato in modo efficace nel contesto accademico. Per essere sfruttato come strumento di supporto alle decisioni, il software dovrebbe essere fornito con le informazioni necessarie alla sua integrazione nei processi decisionali della governance di ateneo. In questo lavoro, presentiamo uno strumento predittivo allo stato dell'arte, fornendone una descrizione funzionale alla sua integrazione, attraverso esempi concreti. Inoltre, proponiamo uno schema semplificato per guidare il ragionamento sul software, che strutturiamo nei seguenti processi: l'apprendimento della macchina, la scelta della rappresentazione dell'abbandono e l'interpretazione dei risultati. Infine, condividiamo alcune considerazioni rivolgendoci ai progettisti dell'educazione e agli *institutional decision-makers* circa la gestione degli interventi ispirati dalla previsione dell'abbandono delle matricole.

Parole chiave: abbandono degli studi; sistema di supporto alle decisioni; modello predittivo.

1. Introduction

Artificial Intelligence (AI) techniques have reached a maturity level which successfully enable their usage in many different contexts, often with surprising results which are covered by the media. In particular, Decision Support Systems (DSS) and predictive models based on Machine Learning (ML) provide tools that help decision making in everyday life and in many cases offer higher performance than human experts. Education is not immune to this revolution; indeed, AI and ML can help to improve in several ways the educational process at different levels.

In this position paper, we consider University level education, and we try to assess the benefits and the limits of DSS and predictive models as tools which can help the governance in some policy making processes. We focus on the problem of University dropout, one of the most complex and adverse observable phenomena in students' careers: a dropout is a potentially devastating event in the life of a student, and it also impacts negatively the University from an economic point of view (Jadrić, Garača, & Čukušić, 2010). Furthermore, a dropout could also be a signal of potential issues in the organization and the quality of the educational proposals.

In a previous work (Del Bonifro, Gabbrielli, Lisanti, & Zingaro, 2020), we developed a tool able to evaluate the risk of quitting an academic course at an early stage, either before the student starts the course or during the first year. We focused on first year students, supported by statistical evidence showing that this time frame is one of the most critical periods for dropout (Barefoot, 2004). However, targeting freshmen means that we could exploit only personal information and previous educational outcomes, e.g. gender, age, high school education, final mark, Additional Learning Requirements (ALR). Thus, so far, we do not have access to some relevant information that could be very helpful for improving the predictive system. As we show in this work, such an improvement can be triggered by the tool integration.

The quality of the ML predictive model depends, by definition, on the quality of the available data used to create it. Here, *quality of data* has several technical meanings and various techniques can be employed to improve it, e.g. by evaluating the features relevance or by transforming them to properly represent the information. It is intuitively clear that when the information contained in the data is not sufficient for the predictive task we are interested in, no predictive model can produce good results. Moreover, when data contain many different information, a proper *features extraction*, i.e. the definition of the appropriate features to represent the data, becomes essential. To tackle such extraction, in Del Bonifro, Gabbrielli, Lisanti, and Zingaro (2020), for a certain cluster of features, we monitored the results by incrementally adding each cluster to the feature set to evaluate their relevance to the task. Often, we seek a trade-off between prediction performance and *data mining* effort. In the dropout case and, more generally, in the education sector such an effort is particularly relevant since:

1. dataset readily available are difficult to find;
2. poor literature exists concerning feature extraction in this application area, that would need competences from different fields;
3. nowadays the educational context is characterized by a massive interaction with digital platforms, thus it becomes crucial to figure out how to integrate such information into decision-making processes (Macfadyen & Dawson, 2010).

To address these issues, we believe that AI-based systems for dropout prediction should be complemented by other software tools which are used by the governance for monitoring the students' careers. These tools support the implementation of suitable interventions, such as specific or support courses for students who exhibit a risky situation. The resulting activities could then be used as features in the predictive system to further improve its quality.

Summarizing, we argue that predictive tools can be used in practice by the University governance to implement several policy making processes and, in particular, to address the dropout problem, provided that they are integrated in a more general software system which allows other forms of control on students' activities. In the remainder of the paper we try to substantiate this claim by providing some concrete examples of applications in the academic context.

2. DSS and predictive models for policy making in University

In the traditional sense, DSS are software that exploit knowledge bases (often formalized with logics) to automatically apply rules and to infer answers which can help users in decision-making. Modern DSS also take advantage of data analysis techniques to extract relevant information from raw data by means of statistical and ML models. Classic ML algorithms or *Neural Networks* are just some of the techniques that have been used to extract relevant information from large knowledge bases.

In Del Bonifro, Gabbrielli, Lisanti, and Zingaro (2020), we addressed the task of dropout prediction by exploiting ML which already proved to be effective in the field of education for evaluating student performance (Li, Lynch, & Barnes, 2018; Márquez-Vera, Morales, & Ventura Soto, 2013). We tackled the challenge of early predicting the dropout by adopting a data-driven approach based on a dataset containing the information about 15,000 freshmen of a major Italian University. Starting from such a dataset, we developed a completely automated learning process creating several models that were able to capture the context in which dropout takes place, hence building a reasonably accurate tool for the dropout prediction. To construct a baseline and assess the challenge of the problem under analysis, we used different well-known classification algorithms: *Linear Discriminant Analysis* (LDA) (Cohen, Cohen, West, & Aiken, 2013), *Support Vector Machine* (SVM) (Chang & Lin, 2011) and *Random Forests* (Breiman, 2001). The software is practically usable and in the approval process for integration into the legacy systems of the academic institution that provided the data.

Most certainly, one of the crucial aspects to take into account when integrating predictive tools and DSS, that can hinder their effective use, concerns privacy and respect for national and European standards, in particular General Data Protection Regulation (GDPR) (https://gdpr.eu/). EU rules require that AI-based software could only be used when they ensure fair and transparent treatment (GDPR, art. 14). In addition, they must provide meaningful information on the logic involved. Although ML techniques provide results of exceptional quality, they often lack human interpretability, providing poor explanation as to why these results were obtained. This *explainability* problem attracts many researchers who use the most diverse mathematical, logical, and algorithmic techniques to seek for a, yet not found, solution. To face such problem, we focus on providing practical information, useful to reveal the steps of the ML process, so as to increase the user awareness of the AI-enhanced system.

3. The ML model cycle for student dropout prediction

Here, we define the life cycle of a predictive model that uses ML techniques to predict the dropout risk. We can streamline the model cycle in three processes: the creation, which includes training and testing, the use, i.e. the act of predicting, and the update.

The creation of the predictive model, in the ML context, implies the execution of an algorithm that *learns* from the examples—that are, in our case, the students' careers—each comprehensive of its outcome. This phase is called *training:* the model recognizes the regularities in the information (*patterns*) w.r.t. the prediction target. Next, we test the model on a small number of examples to verify its ability to *generalize* the prediction, by running the algorithm for careers that were not *experienced* throughout training. In any case, during both training and testing, we know the outcome of the student's career, unlike when we use the model to predict the unknown outcome.

The creation process is highly dependent on how we represent the information we have, i.e. it depends on the properties we decide to include in the data collection. However, it is possible to create different models, either by changing the representation across trainings or by adding new examples. In the following, we refer to each of these new models with the expression *updated model*. In case we need to update the model using the same data involved in the previous training, old data must be properly aligned with the new representation. We discuss this topic in more detail in Section 4.

3.1. A case study of the model cycle

We now present the cycle of creation, use and updating of the predictive model by commenting on an example taken from a practical scenario. We consider students enrolled in a major University and relating to the time interval between the Academic Year (A.Y.) 2016/2017 and 2017/2018.

The creation of the predictive model involves the execution of the training algorithm, fed by the information of the students enrolled for the A.Y. 2016/2017. The data are collected as of 31st of October and contain the actual career outcome of each student for the previous year. An extract of the data, properly anonymized, is shown in Figure 1.

Gender	Age	High school	High school final mark	ALR	Academic school	Economic condition	Course credits	Dropout
M	19	1	100	1	446	2	0	no
F	18	1	100	1	840	5	0	no
M	20	1	85	2	908	4	6	yes

Figure 1. Example of the data provided by the governance of the University.

Each row in Figure 1 corresponds to the description of an individual student and contains information about her condition, divided into columns. For convenience, we refer to the student enrolled in A.Y. 2016/2017 with $S_{2016/2017}$. The first eight columns in Figure 1 are the *features*, while the last column indicates the prediction *target*, i.e. the outcome of the first year (dropout yes/no). By the end of the execution, we obtain the trained predictive model and, concretely, we hold a software able to predict the dropout risk.

Although the prediction is vulnerable to specific types of attacks that could reveal part of the training data, i.e. *model extraction* (Tramèr, Zhang, Juels, & Ristenpart, 2016) and *model inversion* (Fredrikson, Jha, & Ristenpart, 2015), the model does not carry the sensitive values of the features but only a mathematical function that associates them to the outcome. Thanks to this property, and by exploiting the appropriate countermeasures to possible attacks (Zheng, Hu, Fang, & Chengfang Shi, 2019), the software can be run by entities other than those holding the original data, without the need to let circulate students-related information.

The newly created software can be used to predict the outcome of the students' careers enrolled in A.Y. 2017/2018 ($S_{2017/2018}$) and to evaluate the performance of the prediction model by comparing the result with the actual outcome. In the analysis presented in Del Bonifro, Gabbrielli, Lisanti, & Zingaro (2020) the trained model predicts with an accuracy between 65% (SVM algorithm with personal information only) and 90% (LDA algorithm with all available features).

Let us now consider the case of predicting the dropout risk of a generic student S whose career outcome is unknown. At the time of enrollment and whenever S's situation changes, we can predict the risk based on the values of the features available from time to time. In case the algorithm detected a dropout risk, the result can be interpreted as an alarm, indicating the need to design an intervention aimed at reducing such risk, e.g. distribution of a *needs analysis questionnaire* (Sava, 2012), activation of strengthening courses (Pandolfi, Ciampa, Bianchi, Fagnini, & Degl'Innocenti, 2020), provision of study incentives (Covizzi, Vergolini, & Zanini, 2012; Wingate, 2007).

In any case, the very act of intervening in S's career implies a change in her academic history. We can take this change into account by updating the features, either by modifying the existing features or by adding new ones. Once the outcome of S's career is known, we are able to create an updated model that considers the intervention. Over time, the history of each student will enrich the training dataset which, on the one hand, will increase in size and, on the other hand, will comprehend the new variables that code the interventions designed for each case.

4. Representing the student dropout

"Every respectable discipline, that wants to put in the field a reflection on the objects of its own knowledge and its own specific representation, sooner or later is forced to enter into the merits of the matter" (D'Amore, 2005, p. 415).

In this section, we focus on the concept of dropout representation and analyze its evolution w.r.t. the possible uses of the predictive model. With the term representation, we refer to the set of concepts that characterize the dropout of a generic student S and, for convenience, we indicate this set with the capital letter R. We call R_{train} the representation of past information used in the training phase. R_{train} is the set of features $f_1, ..., f_N$ plus the prediction target t: the outcome of S. In Figure 2, we give an example of $R_{train}= \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, t\}$. In the case study of Section 3.1, R_{train} is the representation of $S_{2016/2017}$.

Concretely, the information in our possession take on a tabular form where, for each student, we assign a specific value to every single feature, e.g. "male, 19, scientific high school, 95, no, physics, 5, 0, no". Each set of values, properly encoded, constitutes the model training data (see example in Figure 2).

Label	Description
f_1	Student gender
f_2	Student age
f_3	High school
f_4	High school final mark
f_5	ALR
f_6	Academic school
f_7	Economic condition
f_8	Course credits
t	First year outcome (dropout yes/no)

Figure 2. Information contained in the representation of a student.

We define two other important representations in the life cycle of the predictive model: R_{test} and $R_{predict}$. R_{test} is the dropout representation in the testing phase, composed by the same features as R_{train} and t, while $R_{predict}$ is composed only by the features $f_1,..., f_N$, without prediction target t. Referring to the case study of Section 3.1, R_{test} is the representation of $S_{2017/2018}$, whilst $R_{predict}$ refers to the generic student S, as well as to all those students enrolled in the first year for whom the outcome is unknown.

We continue the case study and try to understand how this relates to R. We assume that the predictive model retrieves a considerable risk of dropout for S and that this enables governance to intervene to reduce such risk. The intervention adds information on the condition of S and implies updating R and, consequently, the model itself. In case we aim to integrate the novel collected information in the predictive model, we need an update that considers the new representation, that includes the intervention. Therefore, we indicate with f_9 the encoded intervention and we build the new representation for the training, composed by the features f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8 and f_9 ($R_{train+intervention}$).

Finally, once the outcome of S's career is known, we can complete the training/update cycle, aligning all the data to $R_{train+intervention}$. We believe that it is reasonable to think that the presence or absence of an intervention influences the student's career, that could result in a possible outcome change.

5. Interpretation of the results of the predictive model

To complete the integration of the prediction tool, here we provide some suggestions that can guide institutional decision makers in the interpretation of the results, both to evaluate the goodness of the model and to design interventions leveraging the information obtained from the prediction.

The prediction for a given career is expressed in symbolic form. More precisely, the output 1 indicates that the predictive model has assigned a probability greater than a fixed threshold[1] for the dropout of that student, 0 otherwise. The interpretation of the prediction may support two different decision-making aspects. First, we interpret to decide the type of intervention to be carried out on the student, i.e. interpretation is the tool that supports the educational institution in the planning phase. Second, we interpret to decide which predictive model to adopt, i.e. interpretation is the tool that allows to compare performances among distinct models and to objectively choose which one to use.

5.1. Interpretation as a tool for intervention on the student's path

The so-called symbolic AI techniques are based on human interpretable representations (Haugeland, 1985) and involve a causal paradigm. Such techniques legitimize the correlation of a particular feature value with a specific outcome, but they require that the causal relationships are known in advance. In our case, to apply these techniques, we should have at least partial knowledge of the factors that explain the dropout phenomenon. On the contrary, the use of sub-symbolic AI techniques, e.g. ML and Deep Learning, generate internal representations of the features that are poorly interpretable by humans and do not allow the correlation between a particular feature value and the outcome of the prediction. In general, symbolic techniques offer greater interpretability; yet, when we face complex phenomena such as the non-completion of a University career, sub-symbolic techniques constitute a less biased approach, which makes possible the discovery of regularities that were otherwise not *visible*.

To use an expression borrowed from the medical field, we could say that the interpretation of the prediction allows to identify those features that represent *risk factors* (Sengen, 2005). A risk factor, in this context, is an indicator of the probability that a particular feature is associated to the dropout condition but does not depend on a cause-effect relationship.

Suppose the model predicts *1* for the student S and the DSS alerts on the need to activate a support intervention. At the moment of analyzing S's situation, we should pay attention not to erroneously attribute a cause-effect relationship between the dropout risk and a specific feature value.

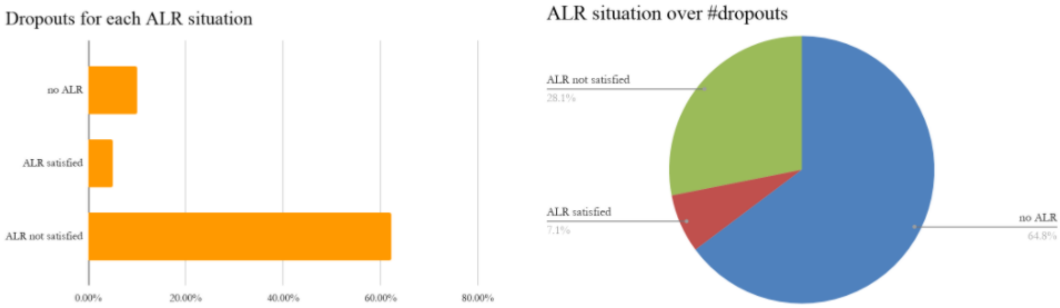

Figure 3. Situation of ALR of students who dropped out.

Let us consider the two graphics in Figure 3, referring to the situation of the ALR for all the $S_{2017-2018}$. On the left, we include the data from the sample of students who have not completed the first year; the percentages refer to the distribution of the three possible ALR situations (lack of ALR, presence of ALR not satisfied, presence of ALR satisfied). On the right, we show, for each of the three ALR situations, the percentage of students who have

[1] The default value for the threshold is set to 50%.

not completed the first year. Interpreting the left graphic, we notice that more than 60% of the students, in the presence of unsatisfied ALR, have dropped out. Intuitively, we could identify this as one of the major causes of dropout. However, this intuition is rejected when we consider the aggregated information on the other graphic: 64.8% of the students who dropped out do not have ALR while only 28.1% of them have ALR not satisfied. In this case, the presence or absence of ALR alone is a good indicator of dropout risk. Both interpretations are reasonable but, when decontextualized, they can lead to the generation of a fallacious cause-effect relationship.

We are able to plan a proper targeted intervention solely after viewing the aggregated information on the whole dataset. This way, we may benefit from a holistic approach to ensure the customization of the intervention, avoiding as much as possible cognitive biases.

5.2. Interpretation as a tool to compare among different predictive models

Suppose now that, after intervening in S's path, we know the outcome of her university career. As we highlight in the Section 4, the novel representation of dropout, which now includes the intervention, makes it necessary to update the model. Thus, two distinct models are available: the one relating to the dropout representation before intervention (M_{old}) and the one relating to the dropout representation which includes intervention ($M_{updated}$). Considering that the model to be integrated in the DSS is unique, we set an objective indicator of the performance and select the model presenting the best measure.

One of the most intuitive performance evaluations can be obtained by comparing predictions with actual outcomes. The *confusion matrix* (CM) is one of the most frequently mean for the assessment of binary classifiers (Sheskin, 2004). As shown in Figure 4, the CM is a double entry table that crosses predictions with actual outcomes.

	Actual outcome: positive	**Actual outcome: negative**
Prediction: positive outcome	This cell records the cases in which a positive result has been predicted and the prediction is consistent with the actual outcome (True Positive, TP).	This cell records the cases in which a positive result has been predicted, but the actual outcome is negative (False Positive, FP, type I error).
Prediction: positive outcome	This cell records the cases in which a negative outcome has been predicted, but the result is positive (False Negative, FN, type II error).	This cell records the cases in which a negative outcome has been predicted and the prediction is consistent with the actual outcome (True Negative, TN).

Figure 4. CM for binary classifiers.

We can calculate common model performance measures[2] by using the values recorded in the CM. In Figure 5, we collect three of those: *accuracy*, *precision* and *recall*.

To clarify the use of CM in the dropout context, we present a concrete example of comparison. We apply the prediction algorithm for both models on a test sample of 100 students. We train M_{old} with information from students enrolled for A.Y. 2016/2017, while we train $M_{updated}$ with both A.Y. 2016/2017 and 2017/2018 information. We assume that the representation for M_{old} is properly aligned with that used for $M_{updated}$. We present the test results of the models in Figure 6 and Figure 7.

[2] wikipedia.org/wiki/Evaluation_of_binary_classifiers

Indicator (performance measure)	Formula	Meaning
Accuracy	$\frac{TP+TN}{TP+FP+FN+TN}$	Calculating the ratio between the cases in which the prediction is consistent with the result and the total cases, a correctness index is obtained.
Precision	$\frac{TP}{TP+FP}$	Calculating the ratio between the cases in which a positive outcome has been predicted correctly and the total cases in which a positive outcome has been predicted, an index of correctness of the prediction, specific for the positive outcome, is obtained.
Recall	$\frac{TP}{TP+FN}$	Calculating the ratio between the cases in which a positive outcome has been correctly predicted and the actual positive cases - which include FNs - it results in a consistency index between prediction and outcome, specific for the positive outcome.

Figure 5. Description of the three most used performance measures for binary classifiers.

	Actual outcome: dropout	**Actual outcome: NON dropout**
Prediction: dropout	TP = 9	FP = 6
Prediction: NON dropout	FN = 4	TN = 81

Figure 6. CM for M_{old}.

	Actual outcome: dropout	**Actual outcome: NON dropout**
Prediction: dropout	TP = 10	FP = 12
Prediction: NON dropout	FN = 3	TN = 75

Figure 7. CM per $M_{updated}$.

	Accuracy	**Precision**	**Recall**
$\mathbf{M_{old}}$	$\frac{9+81}{100}$ = 0.9 (90%)	$\frac{9}{9+6}$ = 0.6 (60%)	$\frac{9}{9+4}$ = 0.69 (69%)
$\mathbf{M_{updated}}$	$\frac{10+75}{100}$ = 0.85 (85%)	$\frac{10}{10+12}$ = 0.45 (45%)	$\frac{10}{10+3}$ = 0.77 (77%)

Figure 8. Comparison between the performance of M_{old} and $M_{updated}$. The cells with grey background identify the best results for each measure.

To select the best of the two models, we need to choose a measure capable of objectively assess the goodness of the single model. In the dropout context, we need to consider that:

1. the phenomenon presents an uneven distribution: the number of students dropping out is much lower than the one of those not dropping out (Del Bonifro, F., Gabbrielli, M., Lisanti, G., & Zingaro, S. P., 2020), and
2. precision and recall reveal two different types of error, type I and type II respectively (Figure 4).

The lack of homogeneity leads us to favor the use of measures that highlight performance on errors rather than correctness. In fact, accuracy for predictive models trained on unbalanced datasets will typically be high, for instance in our example it is high due to TN. Moreover, in the worst case scenario, a measure that performs better on type II errors (corresponding to small FN) leads to intervention policies towards students who are not really at risk of dropping out. Therefore, the recall is the most suitable measure for the performance assessment in the specific case study we are considering. Concretely, it would be reasonable to replace M_{old} with $M_{updated}$ since the latter has a higher recall than the former.

6. Conclusion and future directions

In this paper, we presented an AI-based DSS software to predict the student dropout at University level education, providing a functional description of its integration through practical examples. We organized the description proposing a conceptual framework, to reason about the integrated system, which involves the critical exploration of the following aspects: the cycle of training/updating of the prediction model, the choice for dropout representation, and the possible interpretation of the results.

We believe that increasing the awareness about the ML cycle is crucial to be able to properly manipulate the predictive system. Moreover, we consider plausible the use of our framework to reason on predictive systems with different objectives w.r.t. the dropout. Indeed, we claim that such reference scheme can help to abstract from the technicalities and open the dialogue between those who develop the algorithms and those who use them.

In the future, we plan to increase the effectiveness of the tool as it is used concretely by academic institutions. In addition, we plan to expand the predictive capabilities of the model to include new classification goals, such as transfers and course steps.

Reference list

Barefoot, B. O. (2004). Higher education's revolving door: confronting the problem of student dropout in US colleges and universities. *Open Learning: the Journal of Open, Distance and e-Learning, 19*(1), 9–18.

Breiman, L. (2001). Random forests. *Machine learning, 45*(1), 5–32.

Chang, C. C., & Lin, C. J. (2011). LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST), 2*(3), 1–27.

Cohen, J., Cohen, P., West, S. G., & Aiken, L. S. (2013). *Applied multiple regression/correlation analysis for the behavioral sciences.* New York, NY: Routledge.

Covizzi, I., Vergolini, L., & Zanini, N. (2012). The effects of monetary incentives to University students. *An impact evaluation in Polis, 26*(1), 5–32.

D'Amore, B. (2005). Pipe, cavalli, triangoli e significati. Contributo ad una teoria problematica del significato concettuale, da Frege e Magritte, ai giorni nostri. *L'insegnamento della matematica e delle scienze integrate, 28B*(5), 415–433.

Del Bonifro, F., Gabbrielli, M., Lisanti, G., & Zingaro, S. P. (2020). Student Dropout Prediction. *International Conference on Artificial Intelligence in Education*, 129–140. Cham, CH: Springer.

Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 1322–1333. New York, NY: Association for Computing Machinery.

GDPR. General Data Protection Regulation. https://gdpr.eu/ (ver. 15.12.2020).

Haugeland, J. (1985). *Artificial Intelligence, the Very Idea.* Cambridge, MA: MIT Press.

Jadrić, M., Garača, Ž., & Čukušić, M. (2010). Student dropout analysis with application of data mining methods. *Management: journal of contemporary management issues*, *15*(1), 31–46.

Li, H., Lynch, C. F., & Barnes, T. (2018). Early Prediction of Course Grades: Models and Feature Selection. *arXiv*. https://arxiv.org/ftp/arxiv/papers/1812/1812.00843.pdf (ver. 15.12.2020).

Márquez-Vera, C., Morales, C. R., & Soto, S. V. (2013). Predicting school failure and dropout by using data mining techniques. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, *8*(1), 7–14.

Macfadyen, L. P. & Dawson, S. (2010). Mining LMS data to develop an "early warning system" for educators: a proof of concept. *Computers & education*, *54*(2), 588–599.

Pandolfi, L., Ciampa, A., Bianchi, D., Fagnini, L., & Degl'Innocenti, S. (2020). Design and evaluation of experimental leaving care interventions. *Form@re. Open Journal per la formazione in rete*, *20*(2), 35–51.

Sava, S. (2012). *Needs analysis and programme planning in adult education.* Oplaen, Berlin and Toronto: Barbara Budrich Publisher.

Sengen, J. C. (2005). *Concise Dictionary of Modern Medicine.* New York, NY: McGraw-Hill.

Sheskin, D. (2004). *Handbook of Parametric and Non-parametric Statistical Procedure.* Boca Raton, FL: CRC Press.

Tramer, F., Zhang, F., Juels, A., Reiter, M. K., & Ristenpart, T. (2016). Stealing Machine Learning Models via Prediction APIs. *Proceedings Of The 25Th Usenix Security Symposium*, 601–618. Berkeley, CA: Usenix Association.

Wingate, U. (2007). A framework for transition: supporting "learning to learn" in higher education. *Higher Education Quarterly*, *61*(3), 391–405.

Zheng, H., Ye, Q., Hu, H., Fang, C., & Shi, J. (2019, September). Bdpl: A boundary differentially private layer against machine learning model extraction attacks. *European Symposium on Research in Computer Security*, 66–83. Cham, CH: Springer.