IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 October 2023, accepted 10 November 2023, date of publication 23 November 2023, date of current version 14 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3336596

RESEARCH ARTICLE

Comparing Automated Machine Learning Against an Off-the-Shelf Pattern-Based Classifier in a Class Imbalance Problem: Predicting University Dropout

LEONARDO CAÑETE-SIFUENTES[1], **VICTOR ROBLES**[2], **ERNESTINA MENASALVAS**[2], **AND RAUL MONROY**[1]

[1]School of Engineering and Sciences, Tecnologico de Monterrey, Atizapán de Zaragoza, Estado de México 52926, Mexico
[2]Escuela Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid (UPM), 28660 Madrid, Spain

Corresponding author: Leonardo Cañete-Sifuentes (leonardo.c@tec.mx)

ABSTRACT When facing a classification problem, data science practitioners must search through an armory of methods. Often, practitioners are tempted to use off-the-shelf classifiers, including automated Machine Learning (AutoML) toolboxes; however, stand-alone classifiers are not applicable to every problem and AutoML may be time-consuming raising up environment-ethical issues. To magnify the problem, (commercial) AutoML toolboxes are black and practitioners are not allowed to extend them with new methods to improve their classification performance. Our main objective is to show that an off-the-shelf classifier designed for class imbalance problems can achieve similar performance to an AutoML toolbox. To do so, first, we present the student dropout prediction case study, which most off-the-shelf classifiers find difficult to solve due to the problem's inherent class imbalance. We show that Microsoft Azure AutoML outperforms several popular, stand-alone classifiers. However, multivariate PBC4cip, an off-the-shelf classifier especially designed to deal with class imbalance, yields results that are just as good as Microsoft Azure AutoML, with the advantage that the expensive steps of mechanism selection and tuning are avoided. Our studies show that data science practitioners need to build themselves a taxonomy of classification mechanisms in terms of the properties of the problem to solve. Additionally, AutoML platforms should let scientists modify the armory of classifiers and provide an explanation of both mechanism selection and mechanism tunning so that practitioners learn further lessons.

INDEX TERMS Automated machine learning, feature selection, imbalanced classification models, student drop out, supervised classification.

I. INTRODUCTION

Classification is the task of assigning a class to a given query object. In supervised classification, a mechanism is made to learn this relation by being presented with a collection of labeled examples. In this context, there exist many classification problems, so-called *class imbalance*, where data objects are not distributed uniformly over the classes. These types of problems often involve detecting the presence of something that is not desirable and, luckily, not very common; this includes fraud, HIV, cancer, organ malfunctioning, mental disorders, etc. Class imbalance problems portray one or more classes, called minority, which contain fewer objects than the others, inversely called majority. They pose a problem to a number of classification mechanisms, for they tend to get biased towards the majority class(es).

For domain experts with limited data science knowledge, it is important to use off-the-shelf classifiers; i.e. classifiers

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

that can be trained without the user manually modifying their default parameters. Domain experts may need off-the-shelf classifiers for various reasons, such as the amount of data available in some domains (e.g. biomedical data), the potential shortage of data scientists, or the lengthy interactive process between domain experts and expert data scientists [1]. Since no classifier can achieve good performance on all classes of problems (due to no free-lunch theorem [2]), when randomly selecting an off-the-shelf classifier, we can expect poor classification performance if the selected classifier does not take into account the dataset's characteristics, such as class imbalance.

To enable domain experts to use off-the-shelf classifiers with high classification performance, automated machine learning (AutoML) systems automate the data science pipeline, namely: feature engineering, algorithm selection, and hyperparameter optimization, among other tasks [1]. AutoML systems have shown remarkable classificacion performance; for example, Auto-WEKA [3] showed that the best classifiers for each dataset in their experiments could always be improved by hyperparameter optimization, and that Bayesian optimization outperforms simple search algorithms, such as grid search and random search [4].

There are commercial AutoML solutions such as Microsoft Azure AutoML [5] that provide good usability for beginners [6] and have been successfully applied in problems such as antibiotic resistance prediction [7]. However, commercial AutoML toolboxes are black and they cannot be extended by practitioners to include state-of-the-art classifiers that may improve their performance. Non-commercial automated ML mechanisms, such as those available in Weka or scikit-learn [3], [4], [8] do not share all these problems above mentioned.

In this paper, we present a case study, the student dropout prediction problem, which most off-the-shelf classifiers find difficult to solve due to the class imbalance inherent in it. Student desertion, as defined by the United Nations Organization (1987), is the act of "stopping attending school before the completion of a given stage of education, or at some intermediate time or not end of a school cycle." Student retention aims to predict whether or not a given student is to desert school this is a highly imbalanced classification problem as fortunately, the number of registered students who complete their studies is much larger (around 90%) than those who drop out. To magnify the problem, student retention needs to be carried out in datasets where null values abound, because, at enrollment, students are not usually asked to provide information, like demographics, which later on are considered prominent to explain this phenomenon. Tecnologico de Monterrey has recently released a dataset containing data from 6 cohorts of students [9]: it involves enrollment information (for example, subject), student dropout, and the grades students obtained after the first semester.

The main goal of this study is to compare off-the-shelf classifiers not designed for class imbalance problems, an off-the-shelf classifier designed for class imbalance problems (Multivariate PBC4cip), and a classifier ensemble selected and tuned with AutoML in a real imbalanced dataset. We compare these classifiers in student retention, using Tecnologico de Monterrey's dataset to show how all these algorithms behave in the presence of real data containing null values and a high ratio of class imbalance.

The metrics used to evaluate results were True Positive Rate (TPR) and False Positive Rate (FPR). These ratios are important due to the high imbalance ratio of the problem. It is also worth noting the fact that TPR, or the dropout prediction, is the measure that is of most interest as it denotes the number of predicted students to drop out that really leave their studies.

The main contributions of this paper are:

- A pipeline to preprocess a real dataset containing data of student enrollment, which includes several guidelines for dealing with class imbalance problems.
- A fair comparison of the performance of well-known algorithms in an imbalanced dataset containing enrollment data from 6 different consecutive cohorts running from 2014 to 2020 in a well-known university in Mexico.
- We show that off-the-shelf classifiers that do not consider class imbalance have lower classification performance than an ensemble of classifiers chosen and tuned by automated machine learning (Azure AutoML).
- We show that a single off-the-shelf classifier that considers class imbalance, the contrast pattern-based classifier (Multivariate PBC4cip), has a classification performance comparable to an ensemble of classifiers chosen and tuned by Azure AutoML.

The rest of the paper has been organized as follows. Section II reviews relevant related work; with Section II-A reviewing classifiers for imbalanced data, and Section II-B dedicated to approaches for the dropout problem. Section III is dedicated to show the classification strategy of Multivariate PCB4cip. Section IV describes the dataset and the algorithms that will be compared, and Section describe the methodology that has been used. Section V presents the results and discussion, divided in a comparison of off-the-shelf classifiers not designed for class imbalance against Azure AutoML V-A, and a comparison of Multivariate PBC4cip against Azure AutoML V-B. Finally, Section VI presents the main conclusions and outlook for future work.

II. RELATED WORKS

A. CLASSIFIERS FOR CLASS IMBALANCE PROBLEMS

There exist two main approaches to deal with class imbalance: the data level, and the algorithm level. The data-level approach aims to take a given dataset and get out of it a balanced training dataset, with which one may build an ordinary, supervised classifier. Under-sampling (e.g. [10]) and over-sampling (e.g. [11]) are example mechanisms of this kind. The data-level approach methods share in common the criticism that the altered dataset may be no longer faithful to the original problem observations. By contrast, the algorithm-level approach aims to provide classification mechanisms with the means for dealing with class imbalance without manipulating the input dataset.

[12] describe three common approaches to deal with class imbalance at the algorithm level and their disadvantages. Their discussion is divided into classifiers based on contrast patterns and classifiers that follow other approaches. A contrast pattern is an expression that describes a large proportion of objects from one of the classes when compared to the others; for example [$age > 30$ AND $weight \leq 60$] describes people older than 30 years that weigh more than 60kg. Contrast pattern-based classifiers have the advantage of both providing the rationale behind a classification result, and an understandable decision model. Nowadays, this is of paramount relevance, because for many real-world application domains, such as health, insurance, or finance, it is mandatory to account for any decision result.

Three standard methods for dealing with the class imbalance problem are oversampling, undersampling, and boosting. However, [12] identify disadvantages for each method. The classifiers that use oversampling add synthetic objects with potentially fictitious feature values. Classifiers using undersampling may remove important objects from the sample. Boosting algorithms may extract highly specific patterns that leave parts of the original dataset uncovered.

One contrast pattern-based classifier that is not based on resampling or boosting is PBC4cip [12]. PBC4cip outperforms the SMOTE-TL + LCMine and iCAEP [13] contrast pattern-based classifiers for class imbalance. Furthermore, PBC4cip outperforms other classifiers for class imbalance not based on contrast patterns, such as RUSBoost, Coverage, CTC, and RB-boost.

PBC4cip extracts contrast patterns from a forest of decision trees. The classification performance of PBC4cip has been further improved with Multivariate PBC4cip [14], a contrast pattern-based classifier that extracts multivariate contrast patterns from multivariate decision trees, which allow splits with linear combinations. An example of a multivariate pattern is [$age > 30$ AND $2 * height + 3 * weight > 40$], which describes people older than 30 years old to which the test $2 * height + 3 * weight > 40$ applies.

B. DROP OUT

The student dropout problem is a classification problem that has attracted the attention of researchers and consequently, different reviews and works are available ([15], [16], [17]). We focus in this paper on previous works to predict dropout in Higher Education as this is the challenge of the dataset used in the comparison of algorithms.

In [18], data from 10,196 students between 2013 and 2018 are analyzed. The authors analyze the predictability of dropout both at enrollment and after completing the first semester. The XGBoost classifier obtained the best results with an AUC of 0.92.

In [19], a dataset containing information of 15, 000 students in which the ratio of dropout is 12.3% is analyzed. The authors propose a feature selection strategy and later applied Linear Discriminant Analysis (LDA), Random Forest (RF), and Support Vector Machines (SVM) classifiers. Results reported an accuracy of 0.87, a sensitivity of 0.89, and a specificity of 0.87 all with SVM.

In [20] authors extend the dropout analysis published in [18] and propose to use Deep neural network models and interpretable machine learning techniques such as permutation importance with SHAP to interpret the results. The results reported are 0.77 of AUC for a fully connected deep net.

Important to note is that in this kind of problem it is important to measure the False Positive Rate (students that are predicted to drop that in fact do not abandon) and the ratio of false negatives (students that are predicted to stay when in fact they will abandon studies). It is worth noting that models tend to have a high ratio of false negatives as the majority class is not dropping. Consequently, we will put special attention to the comparison of the algorithms that we will present in how they behave with respect to this issue.

III. MULTIVARIATE PBC4CIP

PBC4cip classifies objects using contrast patterns (CPs), expressions describing a collection of objects. The CPs are extracted from a forest of decision trees during the training phase. The conjunction of tests found from the root to any leaf describes the objects falling into that leaf. This conjunction of tests is a contrast pattern, and PBC4cip extracts CPs by traversing each tree from the root to each leaf. The tests take the form ($x <= v$) or ($x > v$) for numerical features, and ($x = v$) or ($x \neq v$) for nominal features. An example of a contrast pattern is [($scholarship.perc < 0.50$) AND ($region = RM$)], which describes students with less than 50% scholarship that belong to the Monterrey region.

The classification stage of CP-based classifiers is usually based on the support of a pattern for a class. The support of a pattern p for class c, $support(p, c)$, is the proportion of objects from the training dataset of class c that p describes. To classify an object, CP-based classifiers such as CAEP calculate the sum of supports of patterns describing the object for each class. The object is classified with the class with the highest sum of supports. The reason behind this classification strategy is that many patterns with high support for the same class describing an object give evidence that the object belongs to that class. However, [12] point out that in class imbalance problems, many patterns are extracted for the majority class compared to the minority class.

PBC4cip uses a weighted sum of supports to classify objects. To classify an object o, a score is calculated for each class c as shown in Equation 1. The sum of supports of each pattern p that covers the object o is multiplied by a weight w_c

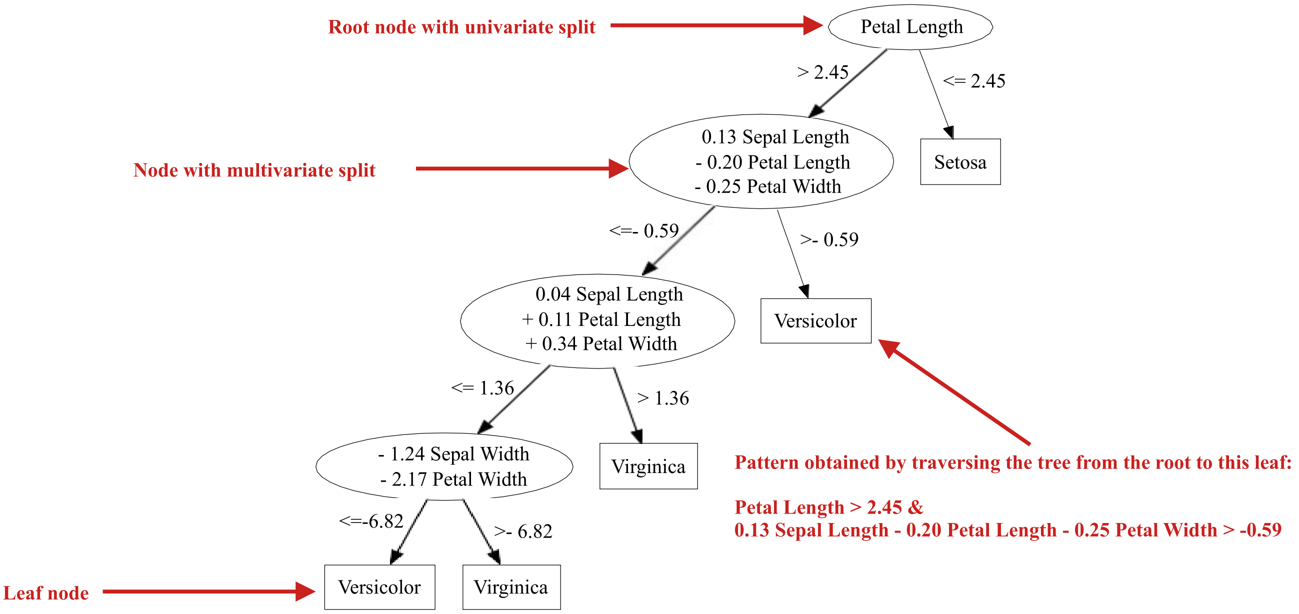

FIGURE 1. Example of an MDT generated with MHLDT for the iris dataset. We give examples of univariate and multivariate splits. Additionally, we show one contrast pattern extracted by traversing the tree from the root to one of the leaves.

that rewards the minority class. The object is classified with the class that maximizes Equation 1.

$$score(o, c) = w_c \sum_{\substack{p \in P \\ p \text{ covers } o}} support(p, c) \qquad (1)$$

The weight w_c, shown in Equation 2, is computed for each class; $|c|$ represents the number of objects of class c, and $|T|$ the number of objects in the training dataset. The smaller the value of c, the higher the value of $(1 - |c|/|T|)$, rewarding the minority class. The denominator is used to normalize the sum of supports in a range [0, 1].

$$w_c = (1 - \frac{|c|}{|T|}) \Big/ \sum_{p \in P} support(p, c) \qquad (2)$$

Multivariate PBC4cip applies the classification strategy of PBC4cip to multivariate contrast patterns, which are contrast patterns where some of the tests contain linear combinations. For example, the multivariate contrast pattern $[(region = RM)$ AND $0.001 * admission.test + 0.0023 * admission.rubric + 0.0010 * english.evaluation > 0.2280]$ describes students from the Monterrey region to which the test involving three numerical features (admission.test, admission.rubric and, english.evaluation) applies. Multivariate CPs are extracted from Multivariate Decision Trees (MDTs), which are DTs where the tests can contain linear combinations.

The original implementation of Multivariate PBC4cip extracts CPs from a forest with the Multi-class Hellinger Linear Decision Tree (MHLDT). We show an example of a tree built with MHLDT in Fig. 1. MHLDT uses a multi-class version of Fisher's Linear Discriminant to generate splits of multiple features. To keep short linear combinations, MHLDT uses Sequential Forward Selection as a feature selection method; this method adds one feature at a time to multivariate splits if the evaluation measure is improved. As an evaluation measure, MHLDT uses Hellinger distance. However, a recent survey on evaluation measures shows that Twoing is better suited to class imbalance problems [21], so the current implementation of Multivariate PBC4cip now uses Twoing.

IV. MATERIALS AND METHODS

A. THE DATASET

The dataset [9] used includes anonymized information related to 77,517 undergraduate students who have enrolled and attended at least one academic term in Higher Education from 2014 to 2020. Among the information categories available in this dataset, we have:

- Sociodemographic information (age, gender, place of origin).
- Enrollment information (program, school, region).
- Academic information related to the student (grade point average obtained in the previous level of studies, current grade point average, current stage; that is, periods completed).
- Information associated with scores on admission tests (PAA, TOEFL, other initial evaluations).
- Academic history (type of school, region, national/international, Higher Education Institution system).
- Student life (participation in sports, cultural, entrepreneurial activities).
- Financial information (type of scholarship, percentage of scholarship).
- Results of the first semester.
- Retention information (whether the student leaves and, in that case, the semester).

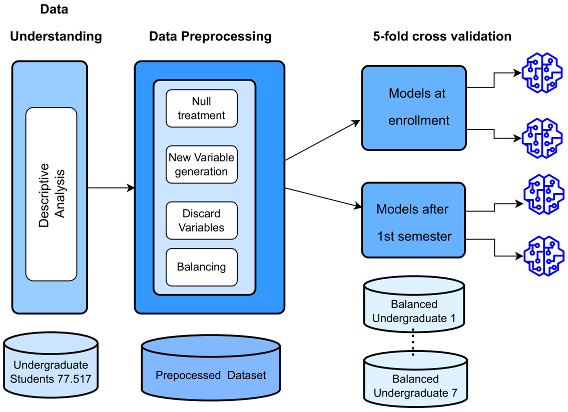

FIGURE 2. Data preprocessing process.

B. METHODOLOGY APPLIED

An adaptation of the CRISP-DM methodology [22] has been applied to understand, explore, and prepare datasets prior to applying the classification algorithms. Once we explored the data, four issues happened to be of special interest: (i) null treatment, (ii) new variable generation, (iii) dataset balancing, and (iv) dataset generation. The process we followed is depicted in Fig. 2.

In what follows we deepen in each of the phases.

C. DATA PREPROCESSING

In Fig. 2, one can observe that first, exploration of the data is performed. The exploration process raised the issue that the dropout ratio is around 9% in all cohorts, which makes the dataset highly imbalanced. This step also highlighted the presence of a high ratio of null values.

Dealing with nulls represents a huge challenge not only owing to the high ratio of null values in some variables but also the different distributions along the data collection period. In fact, information regarding students from the last three cohorts has a lesser proportion of nulls than the previous ones as previously mentioned. Any imputation strategy could consequently produce bias as a much higher amount of null values has to be imputed in the first cohorts. On the other hand, any strategy to balance the data should, in principle, keep the distribution of students along the different cohorts as they were in the raw data.

However, the high number of nulls in many variables, especially in the first cohorts, made us discard them and then calculate models based on data only from the last three cohorts. Despite being a drastic solution, the huge percentage of null values in the discarded cohort makes them unusable to produce models.

Fig. 3 depicts the dataset preparation process. As one can see, after removing the cohorts with high null values, we balance the data using random undersampling. We produce multiple datasets with random undersampling, with dropout ratios ranging from 15% to 45% in steps of 5%. Finally, we divide the dataset for each dropout ratio into a training & test set (90%) and a validation set (10%).

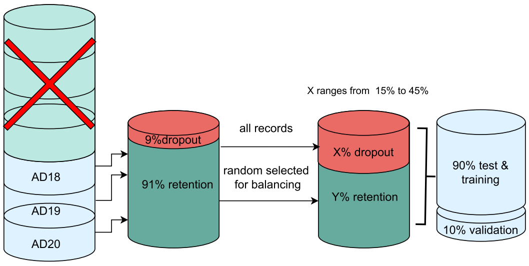

FIGURE 3. Datasets preparation.

Once data is balanced the following variables have been generated:

- some.scholarship: Indicates whether a student has a grant. Possible values: 0 and 1.
- some.activity: Indicates whether a student participated in some activity. Possible values: 0 and 1.

As we have data from the enrollment period, but we also have data from students after the first semester, we decided to calculate models to predict retention at enrollment and after the first semester. In order to generate these models two datasets have been generated:

- DS-enrollment. Contains only information of students up to the enrollment moment.
- DS-behaviour. Contains information up to the enrollment plus behavioural information.

D. MODELLING

Two models will be generated:

- Models for the students once they have enrolled will use DS-enrollment dataset
- Models once the student has followed some semester using the DS-behaviour dataset

To validate the results, we used a validation set with 10% of the data, and 5-fold cross-validation with the training & test set. The metrics used to evaluate results were True Positive Rate (TPR) and False Positive Rate (FPR). These ratios are important in this case due to the high imbalance ratio of the class. It is also worth noting the fact that TPR, or the dropout prediction, is the measure that is of most interest as it denotes the number of predicted students to drop out that really leave their studies.

For the DS-enrollment and the DS-behaviour we obtain classification results with an ensemble chosen by Azure AutoML, Multivariate PBC4cip, and a set of prominent classifiers using their Weka [3] implementations off-the-shelf.

We divide our analysis in two steps:

- First, we compare the ensemble chosen by Azure AutoML against Weka implementations of off-the-shelf decision trees (J48, LMT and FT), tree ensembles (RandomForest and RotationForest), logistic regression

models (Logistic and SimpleLogistic), Naive Bayes, and Multilayer Perceptron.
- Since no classifier from the first step is comparable to Azure AutoML, our second step compares Multivariate PBC4cip only against Azure AutoML.

V. RESULTS AND DISCUSSION

We performed classification on the DS-enrollment and DS-behaviour datasets with the classification models listed in Section IV-D. We divide the comparison of the classifiers in two steps. First, we compare Azure AutoML against a set of prominent off-the-shelf classifiers found in Weka in Section V-A. After showing that no classifier has a comparable performance to Azure, we compare Multivariate PBC4cip against Azure AutoML in Section V-B.

Automated machine learning (AutoML) builds and uses machine learning models in the real world by running systematic processes on raw data and selecting the models that pull the most relevant information from the data. Thus, we have used the next configuration in Azure Machine Learning in order to execute the AutoML:

- Task type: Classification
- Primary metrics: Recall (TPR)
- Explain best model: Enabled
- Validation type: k-fold cross validation
- Number of cross validations: 5
- Percentage test of data (external validation): 10%

For both datasets, the best model chosen by Automated Azure Machine Learning was a VotingEnsemble, formed in turn by several classifier models: XGBoost, LightGBM, RandomForest and ExtremeRandomTrees.

A. COMPARING AZURE AUTOML AGAINST PROMINENT CLASSIFIERS

First, we compare Azure AutoML against the Weka implementation of nine prominent classifiers without the use of hyperparameter optimization (off-the-shelf). We use the default Weka parameters for each classifier to compare the performance of off-the-shelf classifiers against Azure AutoML, which selected an ensemble of classifiers and does hyperparameter optimization.

Fig. 4 shows the results, for the different dropout ratios, in the DS-enrollment dataset. We show the classification results for both the test and validation sets. We notice that the TPR of Azure Ensemble is way higher than that of the other classifiers, reaching a TPR of 63.78% in the test set and 66.69% in the validation set with the original dropout ratio of 8%. On the other hand, the TPR of the other algorithms is below 25% in both test and validation sets with the original dropout ratio. Only when the dropout ratio is balanced to 45% the TPR of some of the Weka classifiers is similar to that of Azure Ensemble.

The best balance of results (TPR vs FPR) for Azure Ensemble is obtained when the dropout ratio is balanced to 35%. Azure Ensemble reaches a TPR of 76.20% with an FPR of 8.12% in the test set and a TPR of 77.52% with an FPR of 7.59% in the validation set. While Azure Ensemble has a higher FPR than the other classifiers with the original dropout ratio, when balanced to 35%, the FPR of Azure Ensemble is comparable to that of the other classifiers (their FPRs are in the range 6.91% − 15.57% in the test set and 6.33% − 15.00% in the validation set).

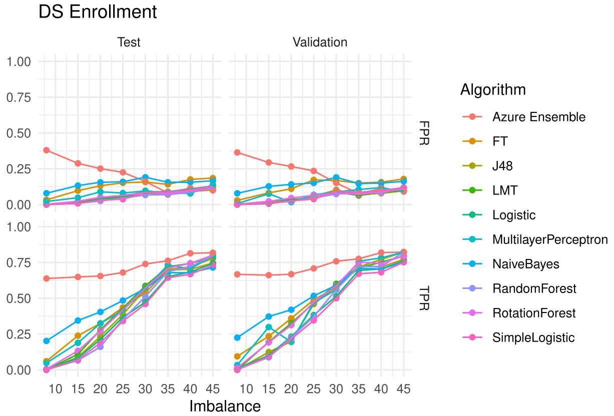

FIGURE 4. True Positive Rate (TPR) and False Positive Rate (FPR) for DS-enrollment for different levels of dropout ratio balance. The left plots correspond to the test set and the right ones to the validation set. The leftmost points in each plot show the results for the original dataset with a dropout ratio of 8%. We can distinguish the results from Azure AutoML ensemble from the Weka classifiers by noting that it has the highest TPR and FPR with a dropout ratio of 8%.

On the other hand, Fig. 5 shows the results for the different dropout percentages in the DS-behaviour dataset. As expected, the results are slightly higher, but the overall behaviour of the classifiers is the same. The main difference is that, when the dropout ratio is balanced to 45% in the validation set, MultilayerPerceptron has a higher TPR (88.93%) than Azure Ensemble (85.91%); however, MultilayerPerceptron also has higher FPR (20.32%) than Azure Ensemble (12.36%).

Although the classification performance of some off-the-shelf classifiers is comparable to that of Azure Ensemble when the dropout ratio is balanced to 45%, we should avoid balancing the dropout ratio if possible. Otherwise, the altered dataset may be no longer faithful to the original problem observations [12], and a classifier trained in such a dataset may not generalize to newer observations.

Azure AutoML manages the best trade-off between TPR and FPR at a lower dropout ratio. So, we now compare Multivariate PBC4cip only against Azure AutoML.

B. COMPARING AZURE AUTOML AGAINST MULTIVARIATE PBC4CIP

We used the default parameters for PBC4cip. By default, PBC4cip extracts contrast patterns from 200 multivariate decision trees and does not filter the contrast patterns. The main parameters of the tree are the number of randomly selected features when splitting a node ($\log_2 features$), the

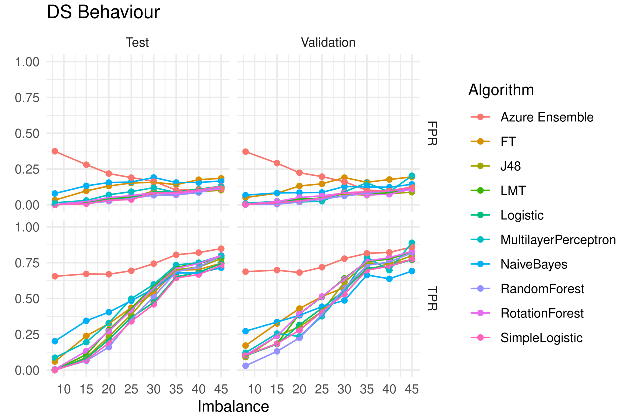

FIGURE 5. True Positive Rate (TPR) and False Positive Rate (FPR) for DS-behaviour for different levels of dropout ratio balance. The left plots correspond to the test set and the right ones to the validation set. The leftmost points in each plot show the results for the original dataset with a dropout ratio of 8%. We can distinguish Azure Ensemble from the Weka classifiers by noting that it has the highest TPR and FPR with a dropout ratio of 8%.

split evaluation function (Twoing), and the minimum number of objects in a leaf (two).

For Azure AutoML, we report the results of the VotingEnsemble as Azure Ensemble, and the results of the best algorithm in the ensemble for each dropout ratio as Azure Best Algorithm.

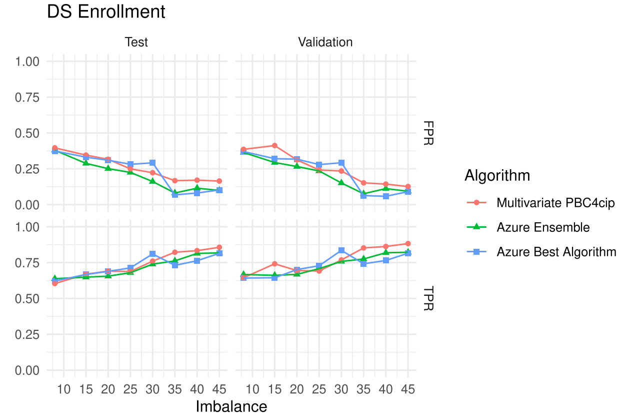

FIGURE 6. True Positive Rate (TPR) and False Positive Rate (FPR) for DS-enrollment for different levels of dropout ratio balance. The left plots correspond to the test set and the right ones to the validation set. The leftmost points in each plot show the results for the original dataset with a dropout ratio of 8%.

Fig. 6 shows the results, for the different dropout ratios, in the DS-enrollment dataset. We show the classification results for both the test and validation sets. As can be seen, the best balance of results (TPR vs FPR) is obtained when the dropout ratio is balanced to 35%. Azure Ensemble reaches a TPR of 76.20% with an FPR of 8.12% in the test set and a TPR of 77.52% with an FPR of 7.59% in the validation set. Azure Best Algorithm reaches a TPR of 73.07% with an FPR of 6.83% in the test set and a TPR of 74.16% with an FPR of 6.33% in the validation set. On the other hand, Multivariate PBC4cip reaches a TPR of 82.21% with an FPR of 16.73% in the test set and a TPR of 85.23% with an FPR of 15.19% in the validation set. The percentages of the final external validation demonstrate the stability of the results obtained.

Fig. 7 shows the results for the different dropout ratios in the DS-behaviour dataset. As expected, the results are slightly higher. In this case, the best results are also obtained when the dropout ratio is balanced to 35%. Azure Ensemble reaches a TPR of 80.53% with an FPR 10.20% in the test set, and a TPR of 81.54% with an FPR of 10.13% in the validation set. Azure Best Algorithm reaches a TPR of 73.89% with an FPR of 6.23% in the test set and a TPR of 74.50% with an FPR 5.61% in the validation set. On the other hand, Multivariate PBC4cip reaches a TPR of 82.84% with an FPR of 15.79% in the test set, and a TPR of 85.57% with an FPR of 15.19% in the validation set.

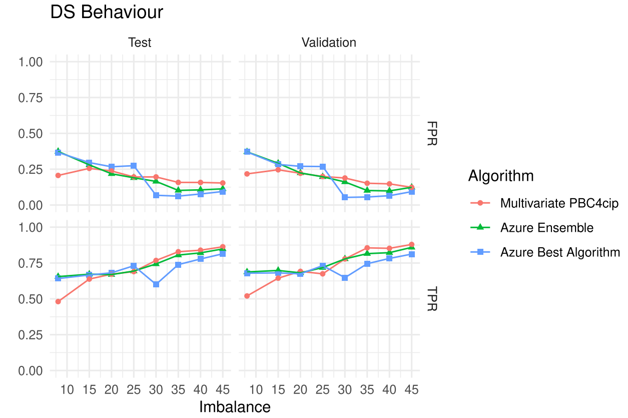

FIGURE 7. True Positive Rate (TPR) and False Positive Rate (FPR) for DS-behaviour for different levels of dropout ratio balance. The left plots correspond to the test set and the right ones to the validation set. The leftmost points in each plot show the results for the original dataset with a dropout ratio of 8%.

In both DS-enrollment and DS-behaviour, we notice a trade-off between Azure AutoML and Multivariate PBC4cip. In general, PBC4cip obtains higher TPR, at the expense of a higher FPR. There are also trade-offs in TPR and FPR between Azure Ensemble and Azure Best Algorithm. Therefore, an administrator can choose one of the three classifiers according to the level of TPR and FPR that they are willing to work with.

We note that Azure Ensemble uses an ensemble of classification models, while Multivariate PBC4cip and Azure Best Algorithm are single classification models. Additionally, Azure AutoML uses a data preprocessing and hyperparameter optimization. A possible way to improve the overall classification results is to include Multivariate PBC4cip in the ensemble used by Azure. However, an advantage of Multivariate PBC4cip is that it provides contrast patterns that can be used to explain classification results. Therefore, it is also worth exploring how to improve contrast pattern-based classifiers for this problem.

TABLE 1. True Positive Rate (TPR) of Multivariate PBC4cip and Azure AutoML for the validation and test sets of DS-Behaviour and DS-Enrollment. We show the TPR for the original dataset with dropout ratio of 8%, and with the modified datasets with dropout ratios balanced from 15% to 45%.

Dataset	Type	Algorithm	8	15	20	25	30	35	40	45
DS-Behaviour	Test	Multivariate PBC4cip	0.48	0.64	0.67	0.69	0.77	0.83	0.84	0.86
DS-Behaviour	Test	Azure Ensemble	0.66	0.67	0.67	0.69	0.74	0.81	0.82	0.85
DS-Behaviour	Test	Azure Best Algorithm	0.64	0.67	0.68	0.73	0.60	0.74	0.78	0.81
DS-Behaviour	Validation	Multivariate PBC4cip	0.52	0.64	0.69	0.67	0.78	0.86	0.85	0.88
DS-Behaviour	Validation	Azure Ensemble	0.69	0.70	0.68	0.72	0.78	0.82	0.82	0.86
DS-Behaviour	Validation	Azure Best Algorithm	0.68	0.68	0.67	0.73	0.65	0.74	0.78	0.81
DS-Enrollment	Test	Multivariate PBC4cip	0.60	0.67	0.69	0.69	0.76	0.82	0.83	0.86
DS-Enrollment	Test	Azure Ensemble	0.64	0.65	0.66	0.68	0.74	0.76	0.81	0.82
DS-Enrollment	Test	Azure Best Algorithm	0.62	0.67	0.69	0.71	0.81	0.73	0.76	0.81
DS-Enrollment	Validation	Multivariate PBC4cip	0.64	0.74	0.69	0.69	0.77	0.85	0.86	0.88
DS-Enrollment	Validation	Azure Ensemble	0.67	0.66	0.67	0.71	0.76	0.78	0.82	0.82
DS-Enrollment	Validation	Azure Best Algorithm	0.64	0.64	0.70	0.73	0.84	0.74	0.77	0.82

VI. CONCLUSION AND FUTURE WORK

Class imbalance problems portray one or more classes, called minority, which contain fewer objects than the others, inversely called majority. They pose a problem to a number of classification mechanisms, for they tend to get biased towards the majority class(es). Several algorithms have been presented to deal with imbalance being Multivariate PBC4CIP the one to show the highest classification performance.

In the last years Automated machine learning (AutoML) methods have shown remarkable performance and helped non-experts to use machine learning off-the-shelf by automating feature engineering, algorithm selection, and hyperparameter optimization, among other tasks. Since no algorithm can achieve good performance on all classes of problems, using AutoML to explore a variety of algorithms and their hyperparameters can help to achieve better classification performance.

In this paper, we compared off-the-shelf classifiers not specifically designed for class imbalance against a classifier ensemble that was chosen with automated machine learning (Azure AutoML). Then, we compare an off-the-shelf classifier (Multivariate PBC4cip) against the classifier ensemble chosen by Azure AutoML.

Dropout is a well-known imbalance problem. For this reason, we compare the classifiers using student retention data from Tecnologico de Monterrey. We compare the classifiers in two moments; first, at the moment of enrollment, and then after the first semester. We created two datasets: the DS-enrollment with only information on students up to the enrollment moment and the DS-behaviour that includes additional behavioural information of the first semester.

In the dataset under study, a high imbalance of the data was observed (dropout ratio of 8%). To deal with class imbalance, we pre-processed the input data. Some lessons to learn from this task include decisions as to getting rid of noisy or incomplete data, to building several datasets, each of which portrays a fixed, different imbalance ratio. Accordingly, we tested the classifiers with the original dataset with the 8% dropout ratio, and with modified datasets with dropout ratios ranging from 15% to 45% in steps of 5% obtained with undersampling. Data was divided into a test and training set with 90% of the data and a validation set with 10% of the data.

The best model chosen by Azure AutoML when optimizing the Recall (TPR) measure was a VotingEnsemble formed in turn by several classifier models: XGBoost, LightGBM, RandomForest, and ExtremeRandomTrees. On the other hand, Multivariate PBC4cip is a standalone classification model.

For the DS-enrollment and DS-behaviour, we gave the True Positive Rate (TPR) and the False Positive Rate (FPR) for each dropout ratio (including the original 8%) in both the test & training, and validation sets. The best balance of TPR and FPR for Azure AutoML and Multivariate PBC4cip is obtained when the dropout ratio is balanced to 35%.

Our results showed that off-the-shelf classifiers not designed for class imbalance problems have lower classification performance than an ensemble of classifiers chosen with Azure AutoML. The standalone contrast pattern-based classifier (Multivariate PBC4cip) has a classification performance comparable to an ensemble of classifiers chosen with Azure AutoML. There is a trade-off in TPR and FPR between Azure AutoML and Multivariate PBC4cip. Multivariate PBC4cip achieves the highest TPR, at the expense of a higher FPR. The trade-off between Azure AutoML and Multivariate PBC4cip guides our future work.

Data science practitioners need to build themselves a taxonomy of classification mechanisms in terms of the properties of the problem to solve; for this case study it is important to acknowledge the class imbalance problem and use a classifier that takes class imbalance into account, such as Multivariate PBC4cip. Additionally, AutoML platforms should let scientist modify the armoury of classifiers and provide an explanation of the mechanism selection so that practitioners learn further lessons.

As future work, first, we propose including Multivariate PBC4cip as a classifier in an AutoML system, since we could obtain a better trade-off between the dropout ratio and the False Positive Rate. A second line of work is to explore improving the classification results of contrast pattern-based classifiers, such as Multivariate PBC4cip, due to their advantage of providing an interpretable model.

TABLE 2. False Positive Rate (FPR) of Multivariate PBC4cip and Azure AutoML for the validation and test sets of DS-Behaviour and DS-Enrollment. We show the TPR for the original dataset with dropout ratio of 8%, and with the modified datasets with dropout ratios balanced from 15% to 45%.

Dataset	Type	Algorithm	8	15	20	25	30	35	40	45
DS-Behaviour	Test	Multivariate PBC4cip	0.21	0.26	0.24	0.20	0.20	0.16	0.16	0.15
DS-Behaviour	Test	Azure Ensemble	0.37	0.28	0.22	0.19	0.16	0.10	0.11	0.11
DS-Behaviour	Test	Azure Best Algorithm	0.37	0.30	0.27	0.27	0.07	0.06	0.08	0.09
DS-Behaviour	Validation	Multivariate PBC4cip	0.22	0.25	0.22	0.20	0.19	0.15	0.15	0.12
DS-Behaviour	Validation	Azure Ensemble	0.37	0.29	0.22	0.20	0.16	0.10	0.10	0.12
DS-Behaviour	Validation	Azure Best Algorithm	0.37	0.28	0.27	0.27	0.05	0.06	0.06	0.09
DS-Enrollment	Test	Multivariate PBC4cip	0.40	0.35	0.32	0.25	0.22	0.17	0.17	0.16
DS-Enrollment	Test	Azure Ensemble	0.38	0.29	0.25	0.22	0.16	0.08	0.11	0.10
DS-Enrollment	Test	Azure Best Algorithm	0.37	0.33	0.31	0.28	0.29	0.07	0.08	0.10
DS-Enrollment	Validation	Multivariate PBC4cip	0.39	0.41	0.31	0.24	0.23	0.15	0.14	0.13
DS-Enrollment	Validation	Azure Ensemble	0.36	0.29	0.27	0.23	0.15	0.08	0.11	0.09
DS-Enrollment	Validation	Azure Best Algorithm	0.37	0.32	0.32	0.28	0.29	0.06	0.06	0.09

APPENDIX. CLASSIFICATION RESULTS: TPR AND FPR

In this section we show the full classification results in tabular format of Multivariate PBC4cip and Azure. The results are show for the DS-enrollment, which Contains only information of students up to the enrollment moment, and for the DS-behaviour, which contains information up to the enrollment plus behavioural information. Both DS-behaviour and DS-Enrollment were divided in a Training & Test set and a Validation set. Additionally, the results are given for the original dataset with an 8% dropout ratio, and for balanced datasets with dropout ratios ranging from 15% to 45%. The True Positive Rate (TPR) is shown in Table 1 and the False Positive Rate (FPR) in Table 2.

ACKNOWLEDGMENT

The authors would like to acknowledge the Living Laboratory and Data Hub of the Institute for the Future of Education, Tecnológico de Monterrey, Mexico, for the data published through the Call "Bringing New Solutions to the Challenges of Predicting and Countering Student Dropout in Higher Education" used in the production of this work.

REFERENCES

[1] J. Waring, C. Lindvall, and R. Umeton, "Automated machine learning: Review of the state-of-the-art and opportunities for healthcare," *Artif. Intell. Med.*, vol. 104, Apr. 2020, Art. no. 101822.

[2] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997, doi: 10.1109/4235.585893.

[3] E. Frank, M. A. Hall, G. Holmes, R. Kirkby, and B. Pfahringer, "WEKA—A machine learning workbench for data mining," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Boston, MA, USA: Springer, 2005, pp. 1305–1314.

[4] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown, "Auto-weka: Automatic model selection and hyperparameter optimization in WEKA," in *Automated Machine Learning—Methods, Systems, Challenges* (The Springer Series on Challenges in Machine Learning), F. Hutter, L. Kotthoff, and J. Vanschoren, Eds. Cham, Switzerland: Springer, 2019, pp. 81–95.

[5] *Cloud Computing Services | Microsoft Azure*, Microsoft, Redmond, Washington, DC, USA, 2023.

[6] T. Pachmann, "An evaluation and comparison of automl solutions: Azure automl and evalml," Hochschule Darmstadt, Darmstadt, Germany, Tech. Rep., 2022 p. 8.

[7] G. Feretzakis, A. Sakagianni, E. Loupelis, D. Kalles, N. Skarmoutsou, M. Martsoukou, C. Christopoulos, M. Lada, S. Petropoulou, A. Velentza, S. Michelidou, R. Chatzikyriakou, and E. Dimitrellos, "Machine learning for antibiotic resistance prediction: A prototype using off-the-shelf techniques and entry-level data to guide empiric antimicrobial therapy," *Healthcare Informat. Res.*, vol. 27, no. 3, pp. 214–221, Jul. 2021.

[8] 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.

[9] T. de Monterrey, "Student dropout dataset," Tecnológico de Monterrey, Monterrey, Mexico, Tech. Rep., 2022.

[10] S. García and F. Herrera, "Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy," *Evol. Comput.*, vol. 17, no. 3, pp. 275–306, Sep. 2009.

[11] A. Gosain and S. Sardana, "Handling class imbalance problem using oversampling techniques: A review," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 79–85.

[12] O. Loyola-González, M. A. Medina-Pérez, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, R. Monroy, and M. García-Borroto, "PBC4cip: A new contrast pattern-based classifier for class imbalance problems," *Knowl.-Based Syst.*, vol. 115, pp. 100–109, Jan. 2017.

[13] X. Zhang, G. Dong, and K. Ramamohanarao, "Information-based classification by aggregating emerging patterns," in *Proc. 2nd Int. Conf. Intell. Data Eng. Automated Learn.*, in Lecture Notes in Computer Science, Hong Kong, vol. 1983, K.-S. Leung, L.-W. Chan, and H. Meng, Eds. Berlin, Germany: Springer, Dec. 2000, pp. 48–53.

[14] L. Cañete-Sifuentes, R. Monroy, M. A. Medina-Pérez, O. Loyola-González, and F. Vera Voronisky, "Classification based on multivariate contrast patterns," *IEEE Access*, vol. 7, pp. 55744–55762, 2019.

[15] C. F. de Oliveira, S. R. Sobral, M. J. Ferreira, and F. Moreira, "How does learning analytics contribute to prevent Students' dropout in higher education: A systematic literature review," *Big Data Cognit. Comput.*, vol. 5, no. 4, p. 64, Nov. 2021.

[16] C. Márquez-Vera, C. R. Morales, and S. V. Soto, "Predicting school failure and dropout by using data mining techniques," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 8, no. 1, pp. 7–14, Feb. 2013.

[17] V. Hegde and P. P. Prageeth, "Higher education Student dropout prediction and analysis through educational data mining," in *Proc. 2nd Int. Conf. Inventive Syst. Control (ICISC)*, Jan. 2018, pp. 694–699.

[18] B. Kiss, M. Nagy, R. Molontay, and B. Csabay, "Predicting dropout using high school and first-semester academic achievement measures," in *Proc. 17th Int. Conf. Emerg. eLearning Technol. Appl. (ICETA)*, Nov. 2019, pp. 383–389.

[19] F. Del Bonifro, M. Gabbrielli, G. Lisanti, and S. P. Zingaro, "Student dropout prediction," in *Proc. Int. Conf. Artif. Intell. Educ.* Cham, Switzerland: Springer, 2020, pp. 129–140.

[20] M. Baranyi, M. Nagy, and R. Molontay, "Interpretable deep learning for university dropout prediction," in *Proc. 21st Annu. Conf. Inf. Technol. Educ.*, Oct. 2020, pp. 13–19.

[21] V. A. S. Hernández, R. Monroy, M. A. Medina-Pérez, O. Loyola-González, and F. Herrera, "A practical tutorial for decision tree induction: Evaluation measures for candidate splits and opportunities," *ACM Comput. Surv.*, vol. 54, no. 1, pp. 1–38, Jan. 2021.

[22] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, "Crisp-dm 1.0: Step-by-step data mining guide," SPSS, CRISP-DM Consortium, NCR Syst. Eng. Copenhagen, DaimlerChrysler AG, SPSS Inc. (USA) and OHRA Verzekeringen en Bank Groep B.V (The Netherlands), Tech. Rep., 2000, p. 13, vol. 9.

LEONARDO CAÑETE-SIFUENTES received the Ph.D. degree in computer science from Tecnológico de Monterrey, Mexico, in 2022. He is currently a Researcher with Tecnológico de Monterrey and the Center of Biotechnology, Universidad Politécnica de Madrid (UPM). His research interests include supervised classification, interpretable classifiers, and large language models.

VICTOR ROBLES received the Ph.D. degree in computer science. From 2012 to 2016, he was the Director of the Universidad Politécnica de Madrid (UPM)'s Higher Technical School of Computer Engineering. During this period, he was the Vice-President of CODDII in Spain (Conference of Directors and Deans in Computer Engineering). Since 2016, he has been a Vice-Rector for Strategy and Digital Transformation with UPM. He is also a Computer Scientist and a Distinguished Professor of computer architecture. He has authored more than 35 research articles, primarily focusing on AI.

ERNESTINA MENASALVAS received the Ph.D. degree in computer science. From 2004 to 2012, she was the Associate Dean of Studies and an Associate Rector for Graduate Studies. She is currently a Databases and Data Mining Professor with Universidad Politécnica de Madrid (UPM). She is also a Computer Scientist. She leads the Data Mining and Data Simulation Group (MIDAS), Center of Biotechnology, UPM. She co-leads the task force on skills in the BDVA-DAIRO. She has participated actively in project development (H2020, FP7, and EIT). She actively participates in EIT-Digital and EIT-Health in special education activities. She has published more than 40 articles. Her research integrates different aspects of data analytics; with the involvement in different real-world problems with special emphasis on health.

RAUL MONROY received the Ph.D. degree in artificial intelligence from The University of Edinburgh, in 1998. Since 1985, he has been with Tecnológico de Monterrey, Mexico, where he is currently a Full Professor in computing and the Founder of the Advanced Artificial Intelligence Research Group. He has held 15 research grants. His research interests include the design and development of novel machine-learning models. He is a fellow of the Mexican Academy of Sciences and the Mexican Academy of Computing. He is also a part of the CONACYT Mexican Research System, Rank 3 (top). He has been the President of the Mexican Society for Artificial Intelligence.