Predicting Student Success Using Data Generated in Traditional Educational Environments

Marian Bucos [1], Bogdan Drăgulescu [2]

[1] *Politehnica University Timisoara, Vasile Parvan Blvd., Timisoara, Romania*

Abstract **– Educational Data Mining (EDM) techniques offer unique opportunities to discover knowledge from data generated in educational environments. These techniques can assist tutors and researchers to predict future trends and behavior of students. This study examines the possibility of only using traditional, already available, course report data, generated over years by tutors, to apply EDM techniques. Based on five algorithms and two cross-validation methods we developed and evaluated five classification models in our experiments to identify the one with the best performance. A time segmentation approach and specific course performance attributes, collected in a classical manner from course reports, were used to determine students' performance. The models developed in this study can be used early in identifying students at risk and allow tutors to improve the academic performance of the students. By following the steps described in this paper other practitioners can revive their old data and use it to gain insight for their classes in the next academic year.**

Keywords **– Classification, Educational Data Mining, predicting student performance, traditional educational environments.**

DOI: 10.18421/TEM73-19
https://dx.doi.org/10.18421/TEM73-19

Corresponding author: Marian Bucos,
Politehnica University Timisoara, Vasile Parvan Blvd., Timisoara, Romania
Email: marian.bucos@cm.upt.ro

Received: 30 June 2018.
Accepted: 11 August 2018.
Published: 27 August 2018.

1. Introduction

Data Mining, also referred to as knowledge discovery from data, is the automatic process of extracting implicit, potentially useful information from data, for better decision-making [1]. The primary function of data mining is the application of specific methods to develop models and discover previously unknown patterns [2]. The increased adoption of Data Mining methods (association, classification, clustering, density estimation, regression, sequential patterns) in education has brought changes in the way tutors, researchers, and educational institutions investigate data to make optimal informed decisions. The implementation of these Data Mining methods for analyzing data available in education defines the field of Educational Data Mining (EDM).

According to Baker and Yacef, the areas that most often attract the attention of EDM researchers include individual learning from educational software, computer-adaptive testing, computer-supported collaborative learning, and the factors that are associated with student failure [3]. On the other hand, the authors summarize the approaches to use with EDM: prediction, clustering, relationship mining, distillation of data for human judgment, and discovery with models. Predicting student's performance is one of the oldest and the most popular application of EDM. Among the techniques that have been used to estimate the student success, the following are mentioned: neural networks, Bayesian networks, rule-based systems, regression, and correlation analysis [4].

Peña Ayala, in his analysis of recent EDM works, observed that student performance modeling is one of the favorite targets of this domain approach. The author identifies several indicators of performance that deserve to be modeled: efficiency, evaluation, achievement, competence, resource consuming, and elapsed time [5].

Higher education institutions are evidently interested in modeling student academic performance to improve quality and efficiency of the traditional

processes [6]. For instance, researchers and tutors could: identify students at risk of failing some of their courses to benefit from an intervention [7]; determine the utility of students' response time in performance prediction [8]; prescribe activities that maximize the knowledge learned as evaluated by expected post-test success [9]; offer individual support according to students' academic skills.

Romero et al. compared several data mining approaches for classifying students' final marks based on the activities carried out in web-based courses [10]. They focused on seven Moodle courses from Cordoba University. All the algorithms were evaluated using stratified tenfold cross-validation. Re-balance preprocessing techniques have also been applied to the original numerical data to evaluate the models. From the conducted experiments, they observed better results for most of the algorithms (17 out of 25) with re-balanced data enabled.

In this paper, we focus on predicting students' performance by only using traditional course report data, gathered over the years by tutors. We aim to prove that old data can have hidden information that can be harnessed by teachers to provide insight for their classes and to provide a good example on how they can accomplish that. Our data is from one distributed examination course, gathered from 2009 to 2017. A binary classification problem is addressed. Our approach to solving this problem includes student performance attributes inside the course, time segmentation of initial data set, different classification algorithms, and two cross-validation methods. The prediction is based on performance attributes (e.g., examination score, activity grade, class attendance) extracted from course reports at the Politehnica University Timisoara, Faculty of Electronics, Telecommunications, and Information Technologies. The output class to be predicted is the academic status that has two possible values: 1 (passed), for students who completed the course, and 0 (failed), for those who need to repeat the course.

The most important aspects that differentiate our study from other works are mentioned in the following. First, we are modeling student academic performance for distributed examination courses, an examination strategy specific to several Romanian universities, including the Politehnica University Timisoara. An approach to data preprocessing that utilizes course report data, already available for tutors and researchers, is described. Next, a data set time segmentation is addressed to enable the algorithms to predict the student course status from early on. Finally, the classification models are evaluated using different cross-validation strategies, thus allowing an analysis of how the results generalize from one year to the next.

The paper is structured as follows. The next section outlines methodology underlying our experiments, the data sets, and the classification models. Section 3 describes the experiments carried out and the results obtained. In Section 4, there is a discussion of this work, and in Section 5, we conclude and discuss future work.

2. Methodology

To build reliable models, Knowledge Discovery and Data Mining process was considered. The process consists of a number of steps, which can be followed in a knowledge discovery project in order to identify patterns in data [11] [12]. The process of knowledge discovery included the following steps: data collection, preprocessing, data mining, and interpretation of the results.

This study aims to reveal student academic performance in distributed examination courses, based on models developed using five classification algorithms implemented in Scikit-learn [13] (Decision Tree CART, Extra Trees Classifier, Random Forest Classifier, Logistic Regression, and C-Support Vector Classification). Each model was evaluated using two cross-validation methods (stratified tenfold cross-validation and leave-one-label-out cross-validation). Students were classified as either 1 (passed) or 0 (failed), in accordance with their performance indicators. The latest version of Scikit-learn (Python package for machine learning), version 0.19, was used to design the experiments for the proposed models.

2.1 Data collection

The data used in this study was obtained from the Faculty of Electronics, Telecommunications, and Information Technologies, Politehnica University Timisoara. The data sets include course records gathered over a period of nine years, from 2009 to 2017, for a distributed examination course, Object Oriented Programming. Therefore, the data has not been gathered expressly for educational data mining purposes. In accordance with the Politehnica University Timisoara regulations, course examination can be achieved through final exam, and distributed examination. For courses with final examination could be established optional midterm examination during the study period, while the final exam is scheduled at the end of semester (final examination period), after completion of the study period. For distributed examination courses, a number of mandatory examinations are provided during the study period. At least two examinations-re-examinations sets must be established to test knowledge, skills and abilities acquired by a student.

Regarding the distributed examination course considered for this study, students participate in 9 optional course sessions, 14 mandatory face-to-face practical activity meetings, and 4 examinations (two of which are re-examinations). For each examination-re-examination set, the greatest value is preserved. In Romanian universities, the marking system is a 10-grading system, where 1 is the lowest grade, and 10 is the highest attainable. For this course, a student needs a mark greater or equal than 5, in average activity mark and each examination mark, to complete the course. The final course mark, at the end of the semester, is calculated as the sum of 50% of the average activity mark and 50% the average examination mark.

For each of the 1077 student records, 43 attributes were collected, such as: name, gender, student membership to advanced study group, study year, grade point average, the number of credits earned in the previous year, attendance for each practical activity meeting (14 attributes), mark for each practical activity meeting (14 attributes), attendance for each examination (4 attributes), mark for each examination (4 attributes), and final course status. The obtained data set consists of student records with a total of 1077 x 43 data points.

The output class to be predicted in our binary classification problem is the academic status that has two possible values: 1 (passed), for students who completed the course, and 0 (failed), for those who need to repeat the course.

2.2 Preprocessing

The data preprocessing refers to any type of processing performed on initial data set to prepare it for applying a data mining technique.

During the data collection step, the data was integrated into a single data set. The data used in this study was taken up from the reports that tutors generate each year for this course. In the initial data set were included 1077 student records. A total of 169 student records was discarded from this data set due to incomplete information (students without class activity and examination data), or duplicate information (the second, the third, or even the fourth registration of a failed student in a course). After removing these student records, the data set contains a total of 908 instances. The data cleaning involved processes like: filling in missing values for some numeric predictor attributes by using the attribute mean; correcting inconsistencies in the data; converting categorical attributes to numeric value due to the requirements of some classification algorithms (C-Support Vector Classification) [14].

The course considered for this study, Object Oriented Programming, takes place in the second year of study. The attributes that were not relevant to the problem were removed from the data sets. For example, the student name is not relevant to the student success evaluation.

Predicting student performance as soon as possible is a major challenge for tutors to better meet the needs of students. To achieve this, four data sets (DS) were derived by including all attributes available after each examination, based on four time segments: the first time segment in week 6 (DS1), the second time segment in week 8 (DS2), the third time segment in week 12 (DS3), and the fourth time segment in week 14 (DS4). The data sets included six common attributes: gender, student membership to advanced study group, study year, grade point average, number of credits earned by each student in the previous year, and final course status. In addition, attributes that describe the performance of students in the course and the number of students from each activity group or course were computed through data aggregation, corresponding to each time segment: average activity mark, the number of attendances in practical activity meetings, average examination mark, the number of examinations, the number of students in one year, and the number of students by study group. The last data set (DS4) was not preserved because it corresponds to the end of the semester.

Each data set contained a total of 908 x 12 data points, with the following attributes: gender (GEN), student membership to advanced study group (MG), study year (SY), grade point average (GPA), number of credits earned in the previous year (CR), number of students in one year (NSY), number of students by study group (NSG), average activity mark (AA), the number of attendances in practical activity meetings (NA), average examination mark (AE), number of examinations (NE), and final course status (SP).

In the data preprocessing step, feature selection based on univariate statistical tests was performed. We computed chi squared statistical test to select the best features from our data sets. During this iterative process those predictor attributes that have the strongest relationship with the output class were selected: student membership to advanced study group, number of credits earned in the previous year, average activity mark, number of attendances in practical activity meetings, average examination mark, and number of examinations. The selected predictor attributes and the output class were given in Table 1. for reference.

Table 1. Attributes used in classification models.

Attribute	Description	Values
CR	number of credits earned in the previous year	numeric value
NA	number of attendances in practical activity meetings	numeric value
NE	number of examinations	numeric value
AA	average activity mark	numeric value
AE	average examination mark	numeric value
MG	student membership to advanced study group	{0=false, 1=true}
SP	final course status	{0=failed, 1=passed}

Data normalization is a good preprocessing practice, required by many classification algorithms. To avoid cases in which predictor attributes have different weight in the decision process standard normalization (z-score) was applied [15]. The normalization was performed using the mean (μ) and the standard deviation (σ) for each attribute. The normalized values were obtained by subtracting the mean of each predictor attribute of the initial data and divide the value by the standard deviation.

$$Xn = (X - \mu) / \sigma \qquad (1)$$

The standard normalization ensures that each attribute has a normal distribution with zero mean and standard deviation equal to one.

In our study, where only a limited amount of data was available, cross-validation methods were used for randomly partitioning of the data into segments of training and test data. Each of the data sets previously obtained on time segments was divided into the training set, part of the data used for learning the models, and the test set, used for evaluating the performance of the classification models [16].

Two cross-validation methods were applied: stratified tenfold cross-validation and leave-one-label-out cross-validation. In the first experiment, stratified tenfold cross-validation method was employed. Stratification is the process of providing for each class equal representation in every fold [17], generally performing better than regular cross-validation in terms of the bias and variance [18]. Using this method, the data was partitioned into ten approximately equal sized segments or folds. By splitting the data sets, each of the ten folds would have roughly 90 records.

Many performance metrics (true positive rate, true negative rate, accuracy, area under the ROC curve, and f-score) were calculated for each model in ten iterations. Each time, one of the ten stratified folds was served as the test set, and the remaining were used as the training set. Thus, for each iteration, a student record is either in the training set or in the testing set. Finally, the average metrics were computed to evaluate the models.

The second experiment involved the use of another method, leave-one-label-out cross-validation. In this approach, each training set is generated by taking all the samples except the ones related to a specific label, while the test set contains the samples left out. For instance, when the case of how the results generalize from one year to another was analyzed, the data sets were partitioned into segments containing only records for one academic year. This way, the data set was partitioned into nine folds, each of them containing only one data group. During nine iterations, the models were trained using student records from eight academic years and evaluated on the left-out year. Similarly to the previous experiment, average metrics were computed using values obtained on metrics in each iteration. Both cross-validation strategies mentioned in this study were implemented using cross-validation and metrics modules from Scikit-learn package [13].

Next, the problem of imbalanced data sets was analyzed. Imbalance in the class distribution may cause a significant deterioration in classification performance [19]. In the first experiment, the class distribution is slightly imbalanced with a percentage of 30% for the minority class. Table 2. shows the distribution of the student academic status.

Table 2. Distribution of the student academic status.

Class	Description	#Students	%Students
1	Passed	639	0.70
0	Failed	269	0.30

The class distribution is quite imbalanced even with the second experiment data sets, with a percentage from 19% to 45% for the minority class. Table 3. summarizes the number of students by year. It shows, for each academic year, the number of students, the number of students who completed the course, the number of students who need to repeat the course, the percentage of passed students, and the percentage of failed students.

In this study, the class imbalance problem has been addressed by assigning weights to classes during classification. The values of the output class have been used to balance weights inversely proportional to class frequencies [13].

Table 3. Data set summary by year.

Year	#Students	%Passed	%Failed
2009	101	0.81	0.19
2010	97	0.77	0.23
2011	74	0.64	0.36
2012	137	0.55	0.45
2013	89	0.55	0.45
2014	87	0.75	0.25
2015	110	0.78	0.22
2016	107	0.75	0.25
2017	106	0.75	0.25

After preprocessing, according to our data set time segmentation approach, the both experiments benefit from three data sets, one for each time segment (DS1, DS2, and DS3), with six predictor attributes. For the first experiment, the data sets were partitioned into ten folds, and for the second experiment, the data sets were divided into nine folds.

2.3 Classification models

Classification refers to the process of learning a model that maps an input data set to a predetermined set of classes. This process consists of two steps, a learning step, where a training set is provided to build the model, and a classification step, where the model is evaluated with a different set of data. According to Tan et al., a classifier is a systematic approach for building classification models from an input data set [20].

To conduct our experiments, we used the most common classification approaches from Scikit-learn: decision tree, logistic regression, ensemble classifiers, and support vector machines. Experiments were performed with: a decision tree (Decision Tree CART) similar to C4.5 decision tree algorithm [21]; an ensemble method (Extra Trees Classifier) that randomizes both attribute and cut-point choice when splitting a node during the construction of the tree [22]; an ensemble classifier (Random Forest Classifier) using the perturb-and-combine technique for decision trees [23]; a linear model (Logistic Regression Classifier, or logit) for classification using a logistic function [24]; an implementation of support vector machine (C-Support Vector Classification) for classification based on libsvm [25].

Table 4. Confusion matrix.

		Predicted	
		Class 1	Class 0
Actual	Class 1	True Positive	False Negative
	Class 0	False Positive	True Negative

The performance of a binary classification model can be evaluated on the number of test records correctly and incorrectly predicted by the model. The following quantities are important for this problem: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). These quantities are associated with a confusion matrix (Table 4.).

In the context of predicting students' academic performance: true positive are students which have correctly been classified as passed; false positive are failed students which have incorrectly been classified as passed; true negative are students which have correctly been classified as failed; false negative are those passed students which have incorrectly been classified as failed. Based on confusion matrix values, qualitative metrics can be defined. In this study, the following metrics were used for evaluating the performance of the classifiers:

- True positive rate (TP rate), also known as sensitivity or recall (Rec), is a proportion of passing students which are predicted to be passed;

$$TP\ rate = TP / (TP + FN) \quad (2)$$

- True negative rate (TN rate), also known as specificity, is a proportion of failed students which are predicted to be failed;

$$TN\ rate = TN / (TN + FP) \quad (3)$$

- Accuracy (Acc) is a proportion of total number of student predictions that were correct;

$$Acc = (TP + TN) / (TP + TN + FP + FN) \quad (4)$$

- F-score (F1) or f-measure is a weighted harmonic average of precision and recall.

$$F1 = 2 \times Pre \times Rec / (Pre + Rec) \quad (5)$$

Another way to evaluate the performance of the classifiers is the area under the ROC curve (AUC). The ROC (Receiver Operating Characteristic) curve is generated by plotting the true positive rate against the false positive rate using different decision thresholds and represents a summary of the qualitative error of a model. The area under the ROC curve corresponds to the probability that a randomly chosen positive instance ranks above a randomly chosen negative instance [26]. For all the classifiers, these metrics were computed using stratified tenfold cross-validation for the first experiment and leave-one-label-out cross-validation for the second experiment.

3. Results

The purpose of the first experiment was to evaluate in which time segment the data set best predicts students' performance. The experiments presented in this study were conducted using Scikit-learn package. The data sets corresponding to the first three examinations (time segments) were used. For each time segment (DS1, DS2, and DS3), five classification algorithms were executed using stratified tenfold cross-validation: Decision Tree CART (DT), Extra Trees Classifier (ET), Random Forest Classifier (RF), Logistic Regression Classifier (logit), and C-Support Vector Classification (SVC).

The prediction results for each time segment and classification algorithm are shown in Table 5., with: true positive rate (TP rate), true negative rate (TN rate), accuracy (Acc), area under the ROC curve (AUC), and f-score (F1). The best results for each data set are highlighted in bold letters.

As the results indicate, the five classifiers performed reasonably well in predicting students' performance, with accuracy ranging from 84% to

86% in week 8 (DS2). Within each time segment, the difference between the highest and the lowest value of the classifiers accuracy varies slightly: 5% for DS1, 2% for DS2, and 4% for DS3. From the obtained results, we can notice that the best accuracy in the first time segment is achieved using Random Forest Classifier, for the second time segment the best accuracy is achieved using any of the following classifiers (Random Forest Classifier, Logistic Regression Classifier, and C-Support Vector Classification), while in the third time segment, Logistic Regression Classifier produces the best result.

Table 5. Classification results using stratified tenfold cross-validation.

Segment	Algorithm	TP rate	TN rate	Acc	AUC	F1
DS1 week 6	DT	0.84	0.56	0.76	0.70	0.83
	ET	0.82	0.62	0.76	0.80	0.83
	RF	**0.85**	0.63	**0.79**	0.83	0.85
	logit	0.72	0.80	0.74	**0.85**	0.80
	SVC	0.72	**0.85**	0.76	**0.85**	0.81
DS2 week 8	DT	0.89	0.74	0.85	0.82	0.89
	ET	0.87	0.75	0.84	0.89	0.88
	RF	**0.90**	0.75	**0.86**	0.90	0.90
	logit	0.87	**0.85**	**0.86**	**0.94**	0.90
	SVC	0.87	0.84	**0.86**	0.93	0.90
DS3 week 12	DT	**0.87**	0.71	0.82	0.79	0.87
	ET	**0.87**	0.72	0.83	0.90	0.87
	RF	**0.87**	0.78	0.85	0.92	0.89
	logit	0.85	0.88	**0.86**	**0.94**	0.89
	SVC	0.82	**0.90**	0.84	0.93	0.88

The best algorithm in predicting passing students (even from the first time segment), with more than 85% in terms of the true positive rate, was Random Forest Classifier. However, the same algorithm offers low performance for the true negative rate. Predicting student academic performance was most often linked to the classification models that will produce the best results in predicting student failure [27]. According to [28], failing to identify students at risk can be as far costlier than incorrectly identifying someone as a failure. C-Support Vector Classification produced the best results in predicting student failure, with values from 85% to 90% in terms of true negative rate. Also, Logistic Regression Classifier offers best value (85%) in predicting student failure with the second time segment.

Experimental results were used in Fig. 1. to illustrate the ROC curves for DS2 and DS3 data sets. The diagrams show that Logistic Regression Classifier and C-Support Vector Classification models are better classifiers for this case. Logistic Regression Classifier produced the best results for AUC metric in every time segment. The highest value was obtained by the DS2 and DS3 data sets, with an AUC of 0.94.

A second experiment was conducted to specifically verify how the models from the first experiment would generalize to an independent year data set. To achieve this, a leave-one-label-out cross-validation method was applied. The initial data sets were divided into nine folds, each of which preserving only one academic year records. Each classifier used eight folds for training, while the validation was performed using the left-out fold.

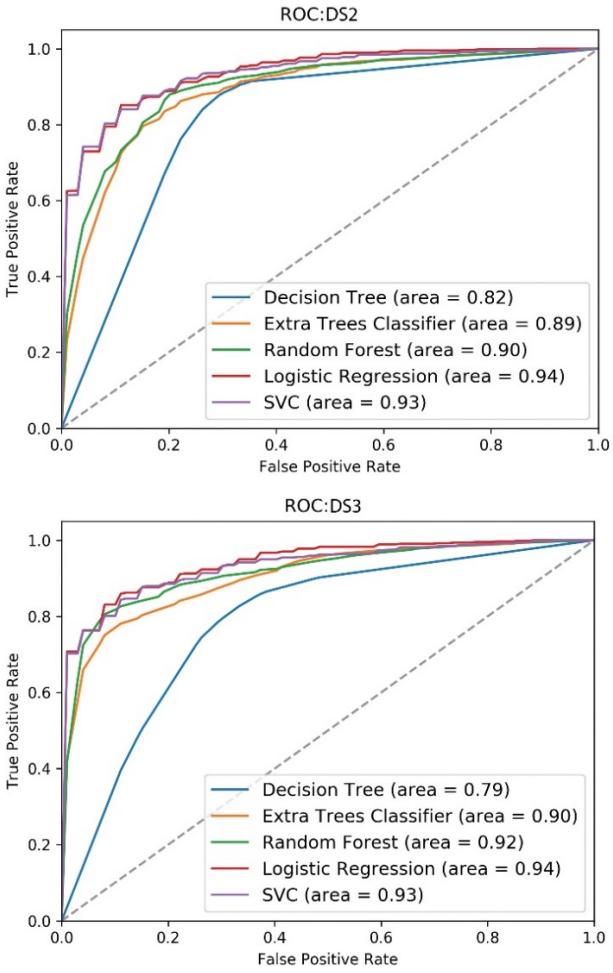

Figure 1. ROC (Receiver Operating Characteristic) curves for classifiers on DS2 and DS3 data sets

The metrics were computed as an overall average of the results across the nine iterations. Table 6. presents the metrics obtained for each classification algorithm and time segment by using leave-one-label-out cross-validation.

The results are similar to those obtained in the first experiment. Once more, the models generated by applying Random Forest Classifier produced the best accuracy with DS1, while in the third time segment Logistic Regression Classifier produces the best result. The highest value (86%) was obtained with the DS2 and DS3 data sets.

Table 6. Classification results using leave-one-label-out cross-validation year partition

Segment	Algorithm	TP rate	TN rate	Acc	AUC	F1
DS1 week 6	DT	0.81	0.54	0.74	0.68	0.81
	ET	0.82	0.54	0.75	0.80	0.82
	RF	**0.85**	0.60	**0.78**	0.80	0.84
	logit	0.71	**0.80**	0.74	**0.86**	0.78
	SVC	0.70	0.79	0.73	0.83	0.78
DS2 week 8	DT	0.89	0.70	0.84	0.80	0.88
	ET	0.88	0.76	0.84	0.90	0.88
	RF	**0.90**	0.77	**0.86**	0.91	0.90
	logit	0.87	**0.87**	**0.86**	**0.95**	0.90
	SVC	0.87	0.85	**0.86**	0.94	0.90
DS3 week 12	DT	0.87	0.69	0.82	0.78	0.86
	ET	**0.89**	0.73	0.84	0.91	0.89
	RF	0.88	0.75	0.84	0.91	0.88
	logit	0.85	0.88	**0.86**	**0.94**	0.89
	SVC	0.82	**0.90**	0.84	0.93	0.88

Regarding true positive rate, the Random Forest Classifier based model offers the highest values, predicting more than 85% of passing students on each data set. As in the previous experiment, the best algorithms in predicting falling students were Logistic Regression Classifier and C-Support Vector Classification. The Logistic Regression Classifier model offer the best results for AUC metric on each time segment, with value of 0.86 for the first time segment, a value of 0.95 in the second time segment, and 0.94 in the third time segment.

4. Discussion

In this paper, we set out to show that data generated in traditional educational environment can be used to predict student academic performance in early stage. The student performance modeling was conducted at the Politehnica University Timisoara with a new type of course, distributed examination course. We consider that in the context of this examination strategy from our university it is possible to build proper classification models to early identify students at risk.

The data used in this study was collected in one course, Object Oriented Programming. The student performance modeling approach described in this study is based on course report data generated over years by tutors. We consider that this kind of data could be summoned by tutors and researchers from higher educational institutions to make informed decisions. The first three examinations, carried out during the semester, allowed us to introduce the course specific data sets (DS1, DS2, and DS3). The fourth examination data set was not preserved due to its correspondence to the end of the semester.

Based on our time segmentation approach, performance attributes were computed through data aggregation after each examination, such as: number of attendances in practical activity meetings, average activity mark, number of examinations sustained, average examination mark. We gave details of how and why such data was collected, cleaned, and normalized. The data collection and aggregation steps are of high importance in the overall problem of developing a prediction model and can take more than half the time needed to develop the models.

Classification models using six attributes were developed in this study. The models were introduced based on different algorithms: Decision Tree CART, Extra Trees Classifier, Random Forest Classifier, Logistic Regression Classifier, and C-Support Vector Classification. These algorithms were chosen from the available ones in Scikit-learn toolkit. Different algorithms may perform better on our data set or on the data sets used by other practitioners that want to implement a prediction model. The point here is to not limit the list of useful algorithms for prediction models to the ones evaluated in this study. It may be regarded as a start but not an exhaustive list. To validate the performance of our classifiers, stratified tenfold cross-validation and leave-one-label-out cross-validation were employed.

In the first experiment, the performance of the classifiers was evaluated based on three metrics: true positive rate, true negative rate, and accuracy. The results of the first experiment are summarized in Table 5. We have shown that by using data generated in traditional educational environment, not collected for prediction models' purposes, classification models can be developed that can predict student academic performance in the early stage with decent metrics. As we expected, the predictive performance of classifiers is changing from one segment to another (more data is available on the activities of students in courses), with significant increases in DS2 against DS1, and minor changes in DS3 against DS2. Even if the classifiers performed reasonably well in predicting students' performance from the first time segment (DS1), the best results relative to the period were offered for the second time segment, in week 8 (DS2). These results are correlated with the structure of our course. Therefore, other practitioners must choose the time segments based on the course structure for which they are building a prediction model.

The second experiment considered the performance of the developed classifiers by using leave-one-label-out cross-validation method. Similarly, to the previous experiment, the performance of the classifiers was evaluated based on the true positive rate, true negative rate, and accuracy. Our goal was to verify how the models would generalize to an independent year data set. We have shown that leave-one-label-out cross-validation

method could be used to validate models against time-based folds and, therefore, evaluate how the model generalize over the years. It is mandatory for a practitioner to evaluate the models for quality over the years. Some tested algorithms presented a significant drop in performance from tenfold cross-validation to leave-one-label-out, for example SVC for the first time segment DS1 on the metric TN rate it had a drop of 0.06. In the second experiment, leave-one-label-out cross-validation was employed. To implement the leave-one-label-out approach, all but one of the available academic years were used in the training set and the left-out year records were used to validate the results. The results obtained for each classification algorithm and time segment by using leave-one-label-out cross-validation are shown in Table 6.

After measuring the metrics of the classifiers by using both stratified cross-validation and leave-one-label-out cross-validation, we can conclude that Logistic Regression Classifier produced best values for this classification problem. While this study extends our research in predicting student academic performance, focusing on applying data mining techniques on traditional report data, we acknowledge several limitations, some already mentioned in the above discussions. A major limitation is the use of data generated within one course and, therefore, the possibility of using the models for other courses.

To mix data and build a general model for multiple courses the following problems must be considered: how the models could be validated for courses that have a different structure, courses at the same university but on different academic tracks, or courses from different universities. If the practitioner has a well organize data from the previous years, the more logical approach is to develop the models for each individual course.

In addition, a downside of the proposed approach is that we mainly rely on student performance attributes. What to do if there is data collected by using electronic means. In this study, we focused on only using old data collected traditionally to prove that it can be used to build reliable prediction models. If the practitioner has access to data collected by an e-learning platform, it is logical to mine the data and produce extra attributes for the training sets that can improve the performance of the models.

5. Conclusions

This paper describes a study in predicting student academic performance by using only report data generated over years in one distributed examination course from Politehnica University Timisoara.

The experimental results have shown that students at risk can be identified in early stage even from performance course data that has not been gathered expressly for educational data mining purposes. From our results, we found that Logistic Regression Classifier is the best algorithm for this classification problem, offering good values in terms of accuracy, true negative rate, and true positive rate. However, other classifiers are performing well, even from the first time segment. Although we only used a limited amount of data, we have shown that classifiers performance evaluation may be conducted through cross-validation (stratified tenfold cross-validation, leave-one-label-out cross-validation).

Properly used, the classification models developed in this study could help us to identify students at risk of failing to benefit from an intervention, or to design student's activities according to their skills or interests. By following the steps described in this paper, practitioners that have not used their old data can develop models to help them get insight into their students' performance.

Directions for further research emerged during the conduct of this study. The most important direction would be to extract data, regarding this course or some other courses, from our university Learning Management System, based on Moodle.

References

[1]. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

[2]. Maimon, O., & Rokach, L. (2009). Introduction to knowledge discovery and data mining. In *Data Mining and Knowledge Discovery Handbook* (pp. 1-15). Springer, Boston, MA.

[3]. Baker, R. S., & Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. *JEDM| Journal of Educational Data Mining*, *1*(1), 3-17.

[4]. Romero, C., & Ventura, S. (2010). Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *40*(6), 601-618.

[5]. Peña-Ayala, A. (2014). Educational data mining: A survey and a data mining-based analysis of recent works. *Expert systems with applications*, *41*(4), 1432-1462.

[6]. Delavari, N., Phon-Amnuaisuk, S., & Beikzadeh, M. R. (2008). Data mining application in higher learning institutions. *Informatics in Education-International Journal*, *7*, 31-54.

[7]. Vihavainen, A., Luukkainen, M., & Kurhila, J. (2013, July). Using students' programming behavior to predict success in an introductory mathematics course. In *Educational Data Mining 2013*.

[8]. Xiong, X., Pardos, Z., & Heffernan, N. (2011). An analysis of response time data for improving student performance prediction. KDD 2011 Workshop: Knowledge Discovery in Educational Data.

[9]. Yudelson, M., & Brunskill, E. (2012). Policy building -- an extension to user modeling. Proceedings of the fifth International Conference on Educational Data Mining, (pp. 188-191).

[10]. Romero, C., Ventura, S., Espejo, P., & Hervas, C. (2008). Data mining algorithms to classify students. Proceedings of Educational Data Mining 2008: 1st International Conference on Educational Data Mining, (pp. 8-17).

[11]. Fayyad, U. (1996). Data mining and knowledge discovery: making sense out of data. *IEEE Expert*, *11*(5), 20–25.

[12]. Kurgan, L., & Musilek, P. (2006). A survey of knowledge discovery and data mining process models. *The Knowledge Engineering Review*, *21*(1), 1-24.

[13]. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, *12*(Oct), 2825-2830.

[14]. Lee, N., & Kim, J. (2010). Conversion of categorical variables into numerical variables via Bayesian network classifiers for binary classifications. *Computational Statistics & Data Analysis*, *54*(5), 1247-1265.

[15]. Han, J., Kamber, M., & Pei, J. (2011). Data Mining: concepts and techniques (3Ed.). Morgan Kaufmann Publishers.

[16]. Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys*, *4*, 40-79.

[17]. Hens, A. B., & Tiwari, M. K. (2012). Computational time reduction for credit scoring: An integrated approach based on support vector machine and stratified sampling method. *Expert Systems with Applications*, *39*(8), 6774-6781.

[18]. Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *International Joint Conference on Artificial Intelligence*, *14*(2), 1137-1143.

[19]. Van Hulse, J., Khoshgoftaar, T. M., & Napolitano, A. (2007, June). Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine learning* (pp. 935-942). ACM.

[20]. Tan, P. N. (2007). *Introduction to data mining*. Pearson Education India.

[21]. Breiman, L., Friedman, J., Stone, C., & Olshen, R. (1984). Classification and regression trees. CRC press.

[22]. Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3-42.

[23]. Breiman, L. (2001). Random forest. *Machine Learning*, *45*(1), 5-32.

[24]. Yu, H., Huang, F., & Lin, C. (2011). Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, *85*(1-2), 41-75.

[25]. Chang, C., & Lin, C. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 1-27.

[26]. Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874.

[27]. Márquez-Vera, C., Cano, A., Romero, C., & Ventura, S. (2013). Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data. *Applied Intelligence*, *38*(3), 315-330.

[28]. Aguiar, E., Chawla, N. V., Brockman, J., Ambrose, G. A., & Goodrich, V. (2014, March). Engagement vs performance: using electronic portfolios to predict first semester engineering student retention. In *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge* (pp. 103-112). ACM.