A decision support system for predicting students' performance

Ioannis E. Livieris[1], Tassos A. Mikropoulos[2], Panagiotis Pintelas[3]
livieris@gmail.com, amikrop@uoi.gr, pintelas@upatras.gr

[1] Technical Educational Institute of Peloponnese, Department of Accounting & Finance, Greece
[2] The Educational Approaches to Virtual Reality Technologies Lab, University of Ioannina, Greece
[3] Department of Mathematics, University of Patras, Greece

Abstract. Educational data mining is an emerging research field concerned with developing methods for exploring the unique types of data that come from educational context. These data allow the educational stakeholders to discover new, interesting and valuable knowledge about students. In this paper, we present a new user-friendly decision support tool for predicting students' performance concerning the final examinations of a school year. Our proposed tool is based on a hybrid predicting system incorporating a number of possible machine learning methods and achieves better performance than any examined single learning algorithm. Furthermore, significant advantages of the presented tool are that it has a simple interface and it can be deployed in any platform under any operating system. Our objective is that this work may be used to support student admission procedures and strengthen the service system in educational institutions.

Keywords: Educational data mining, machine learning, decision support tool, prediction, student evaluation system

Introduction

Educational Data Mining (EDM) constitutes a new research field, which gained popularity in the modern educational era because of its potential to improve the quality of the educational institutions and system. During the last decade, this area of research field has grown exponentially, spurred by the fact that it enables all educational stakeholders to discover new, interesting and useful knowledge about students and potentially improve some aspects of the quality of education.

The importance of EDM is founded on the fact that it allows educators and researchers to extract useful conclusions from sophisticated and complicated questions. More specifically, while traditional database queries can only answer questions such as "find the students with poor performance", data mining can provide answers to more abstract questions like "find the students who will exhibit poor performance" (Livieris et al., 2016). Hence, the application of EDM is mainly concentrated on the development of accurate models that predict student characteristics and performances in order to improve learning experiences. The accurate prediction of students' academic performance is important for making admission decisions as well as providing better educational services (Baker & Inventado, 2014; Mohamad & Tasir, 2013; Romero & Ventura, 2010).

Secondary education in Greece is a two-tied system; the first three years cover general education followed by another three years of senior secondary education. Hence, the three years of higher secondary education, which is also known as Lyceum is a significant and decisive factor in the life of any student for opting desired subjects of study in higher education. In fact, Lyceum acts like a bridge between school education and higher learning specializations that are offered by universities and higher technological educational institutes. Therefore, the ability to predict the students' performance with high accuracy in many stages of the academic period is considered essential for an educator for identifying slow learners and distinguishing "weak" students who are likely to have low achievements. For the prediction of the students' performance, the educators can utilize the students' oral and written examinations and their grades in a small number of evaluation tests as powerful tools for decision making. Subsequently, the prediction results can be validated by lectures in order to specify the most suitable interventions for each group of students and provide them with further assistance tailored to their needs. Furthermore, accurate identification of weak students is one way to provide better educational services by limiting the students who are likely to have low achievements or even guiding them to follow technical education. Hence, developing an accurate prediction tool is very important for an educator and for educational institutions, in general.

During the last decade, much research has been devoted to develop an efficient and accurate prediction model based on a classifier for predicting the student's future academic performance. Nevertheless, the development of such prediction model is a very attractive and challenging task (Baker & Yacef, 2009, Romero & Ventura, 2007; 2010; Romero et al., 2010 and references therein). The primary reason is that datasets from this domain skewed class distribution in which most cases are usually located to the one class (Kotsiantis, 2012; Kotsiantis, Pierrakeas & Pintelas, 2003; 2004). Thus, a classifier induced from an imbalanced dataset has typically a low error rate at the majority class and an unacceptable error rate for the minority classes. Moreover, searching for the best prediction method is still in progress which makes the decision of the selection of a particular learning algorithm for a specific problem, a very complicated problem. To the best of our knowledge, a good alternative for choosing only one method is to create a hybrid forecasting system incorporating a number of possible machine learning methods as components. Thus, the concept of combining learning algorithms has been proposed as a new direction for improving the performance of individual classifiers and obtaining more accurate and efficient predictions.

In this work, we present the design, implementation and application of a new decision support tool for predicting students' performance at the final examination in the discipline of Mathematics. We have implemented a hybrid system that combines the predictions of learning algorithms using simple voting methodology and achieves better performance than any simple method. Furthermore, the proposed hybrid model has been incorporated in a user-friendly software tool for the prediction of students' performance in order to make this task easier for educators to early identify weak students with learning problems. A significant advantage of the presented tool is that it can be deployed in any platform, under any operating system. Our objective is that this work could be used as a reference for decision making in the admission process and to provide better educational services by offering customized assistance according to students' predicted performance.

The paper is organized as follows: The next section presents some elementary machine learning definitions and a more detailed description of the utilized techniques and algorithms in our framework. The following section reviews the related work of other researchers in the area of machine learning algorithms for prediction and classification in education. The next section presents the educational dataset utilized in our study and a series of tests in order to examine the accuracy of each learning algorithm in the specific dataset. The following section presents our software tool and its main features. Finally, the last section discusses the conclusions and some future research directions.

A review of supervised machine learning techniques

Supervised machine learning is a special case of data mining that concerns the process of predicting unknown attribute values from a given set of known attribute values (Mitchell, 1997). For this purpose, a large number of techniques and algorithms have been developed based on artificial intelligence and statistics. In the rest of this section, we present the most popular classes of classification algorithms, which include Bayes classifier, artificial neural networks, rules induction algorithms, instance-base classifiers, decision trees and support vector machines.

A Bayesian network is structured as a combination of a directed acyclic graph of nodes and links and a set of conditional probability tables (Jensen, 1996; Mitchell, 1997). Each node in the graph is associated with a feature whereby the links between nodes represent the relationships between them and the strength of the links is determined by conditional probability tables. More analytically, each node in the network has an associated probability table that describes the conditional probability distribution of that node given its parents nodes. If a node has one or more parents the probability distribution is a conditional distribution, where the probability of each attribute depends on the values of the parents while in case a node has no parents the probability distribution is unconditional. Using a suitable training method, one can induce the structure of the Bayesian network from a given training set (Jensen, 1996). The classifier based on this network and on the given set of attributes X_1, X_2, …, X_n returns the label c that maximizes the posterior probability $p(c \mid X_1, X_2, \ldots, X_n)$.

Artificial Neural Networks (ANNs) are parallel computational models comprised of densely interconnected, adaptive processing units, characterized by an inherent propensity for learning from experience and also discovering new knowledge. Classification with a neural network takes place in two distinct phases. Firstly, the network is trained on a set of paired data to determine the input-output mapping by fixing the weights of the connections between neurons and then, the network is used to determine the classifications of a new set of data (Bishop, 1995; Haykin, 1994; Rumelhart, Hinton & Williams, 1986). The excellent capability of self-learning and self-adapting of ANNs has established them as vital components of many systems. They are considered as a powerful tool for pattern classification. Thus, they have been successfully utilized to tackle difficult real-world problems (Bishop, 1995; Haykin, 1994) and are often found to be more efficient and more accurate than other classification techniques (Lerner et al., 1999; Livieris, Drakopoulou & Pintelas, 2012). Nevertheless, the main disadvantage of ANNs is the computational cost since the process of building and training the network model can be especially time-intensive.

In rule induction systems, a decision rule algorithm creates a set of rules representing the profile of each category defined as a sequence of Boolean clauses linked by logical AND and OR operators that together imply membership in a particular class (Furnkranz, 1997). The primary goals are to identify strong rules discovered in databases using different measures of interestingness and to construct the smallest rule-set that is consistent with the training data. During the classification phase, the left hand sides of the rules are applied sequentially until one of them evaluates to true, and then the implied class label from the right hand side of the rule is offered as the class prediction.

Instance-based learning algorithms stand for a family of machine learning algorithms which delay the induction or generalization process until classification is performed (Aha, 1997; Aha, Kibler & Albert, 1991; Mitchell, 1997). These algorithms are developed from the need to perform discriminant analysis when reliable parametric estimates of probability densities are unknown or difficult to determine. One of the most important characteristics of this class of algorithms is the absence of the initial classifier training phase since they do not build any classification model or any abstraction from the data. However, these algorithms use the whole training data as part of the classifier to classify unseen instances (Aha et al., 1991). This kind of classifiers evolve around a classic learning algorithm called k-Nearest-Neighbor (k-NN) which is based on the principle that the examples within a dataset will generally exist in close proximity with other examples that have similar properties. The main advantages of the k-NN-based classification method is its easiness and simplicity of implementation and the fact that it provides good generalization results during classification assigned to multiple categories.

Decision trees are among the most widely and broadly used algorithms for supervised classification learning. Their recursive construction creates a model based on a tree structure using a set of training examples and aim in separating examples belonging to separate categories (Kohavi & Quinlan, 1999). Decision trees can be represented as influence diagrams, focusing on relationships between particular nodes. Each node in a decision tree represents an attribute of an instance, with branches representing possible values connecting features. A leaf representing the class terminates a series of nodes and branches. The determination of the class of an instance is a matter of tracing the path of nodes and branches to the terminating leaf. Thus, the created model is readily interpretable since it can graphically describe the decisions to be made, the events that may occur, and the outcomes associated with combinations of decisions and events. Furthermore, an additional advantage of the decision trees is that they do not impose statistical assumptions on data distribution. More information about the existing work in decision trees can be found in Mitchell (1997), Murthy (1998), and Quinlan (1993).

The Support Vector Machines (SVM) are a group of supervised learning methods established as part of the most precise discriminatory methods used in classification. They represent an extension to nonlinear models of the generalized portrait algorithm of Vapnik (Vapnik, 1995) which is based on structural risk minimization, an inductive principle of use in machine learning. The training procedure is based on the set of labeled training examples, which are processed during the quadratic programming to find the hyperplane separating optimally examples from different categories. However, in most real-world problems there exists no such hyperplane that successfully separates the instances in the training set since they involve non-separable data. Hence, an elegant way to address this inseparability problem is to map the data into a higher-dimensional space and define a separating hyperplane there. This higher-dimensional space is called the feature space, as opposed to the input space occupied by the training instances. With an appropriately chosen feature space of sufficient dimensionality, any consistent training set can be made separable.

Ensemble of classifiers

In the last two decades, in the area of machine learning there has been proposed a new direction for improving the performance of single classifiers by combining the predictions of a variety of classifiers. More specifically, an ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way to classify new instances (Kotsiantis, 2007). The basic idea of ensemble methodology is the combination of a set of models, each of which solves the same original task, in order to obtain a better composite global model, with more accurate and reliable estimates or decisions than can be obtained from using a single model (Rokach, 2010).

Several methods have been proposed for the creation of an ensemble of classifiers. The most common and widely used method is to use a variety of algorithms on the training data and combine their predictions utilizing a voting scheme. An advantage of this technique is to exploit the diversity of the errors of the learned models by utilizing different learning algorithms, which vary in their method of search and/or representation (Merz, 1997; Merz, 1999). Another methodology to combine a generated diverse set of models is called stacked generalization or simply Stacking. Stacking combines multiple classifiers to induce a higher-level classifier with improved performance. Its basic idea is to consider the voting step as a separate classification problem, whose input is the vector of the responses of the base classifiers. Simple voting predicts the most frequently predicted class based on the number of predictions for each class in the input, while in contrast stacking replaces this with a new classifier (Wolpert, 1992). The matrix containing the predictions of the base learners as predictors and the true class for each training case is called the meta-dataset while the classifier training on this matrix is called meta-classifier. In the grading methodology the meta-level classifier predicts whether the base-level classifier is to be trusted i.e. whether its prediction is correct. Only the base-level classifiers that are predicted to be correct are taken and their predictions combined by summing up the probability distributions predicted. The base-level attributes are also utilized as meta-level attributes, while the meta-level class values are 1 (correct) and 0 (incorrect). More information about ensembles of classifiers can be found in Dietterich (2001), Kuncheva (2014), Rokach (2010) and the references therein.

Literature review

During the past decade, the application of several data mining techniques is becoming very popular in the modern educational era, enabling the development of efficient and accurate models that predict students' academic performance.

Independently, Kabra & Bichkar (2011), Baradwaj & Pal (2011) and Anju & Robin (2013) have conducted surveys on decision trees classification algorithms to predict student academic performance and extract knowledge that describes their performance in the examinations. Kotsiantis et al. (2003; 2004) described models to predict students' future behavior for a distant learning course in Hellenic Open University using grades in written assignments, attendance and students' demographics as attributes. Based on previous works Kotsiantis (2012) developed a prototype decision support system based on regression techniques for predicting students' future grades. Cortez & Silva (2008) conducted a performance study on students selected from two secondary schools in two core disciplines (Mathematics and Portuguese). They applied four classification algorithms in order to identify the students who are likely to fail in the classes. Based on their results, the authors concluded that a good predictive accuracy can be achieved, provided that the first and/or second school period grades are available. Oladokun, Adebanjo & Charles-Owaba (2008) utilized a neural network classifier to predict the performance of a candidate being considered for admission into university. The results indicated that the model is able to correctly predict the performance of more than 70% of the prospective students. Moreover, they developed another neural network model to predict the students' final achievement and categorized them into two groups. Their preliminary results showed that an accurate prediction is possible at an early stage, more specifically at the third week of the 10-week course.

Baker & Yacef (2009), Romero & Ventura (2007; 2010) and Romero et al. (2010) have provided excellent reviews of how EDM develops techniques and approaches to understand the learning process as well as the major trends in EDM research. In their reviews, they presented how EDM seeks to discover new insights into learning with new tools and techniques, so that those insights impact the activity or practitioners in primary, secondary and higher education, as well as corporate learning. Furthermore, they described the process of mining learning data step-by-step, as well as how to apply the data mining techniques such as statistics, visualization, classification, clustering and association rule mining.

Motivated by the previous works, Livieris et al. (2012) developed a user-friendly software tool that is based on a neural network classifier for predicting the students' performance in the discipline of Mathematics of the first year of Lyceum. Based on their numerical experiments, they concluded that the neural networks exhibit more consistent behavior and illustrate better classification results than the other classifiers. On the basis of this idea, Chen & Do (2014) presented a comparative study with its main objective to investigate the prediction ability of the neural networks for students' performance prediction using as input variables previous exam results, students' gender and other demographics attributes. In more recent works, Pandey & Taruna (2014) studied the performance of several classifiers for automatically identifying weak students and proposed a multilevel classification model. Moreover, they incorporated pre-processing techniques such as resample filter as well as removing the misclassified instances from the initial classifier in order to enhance the classification accuracy of the model.

Methodology

The aim of this study is to develop a decision support tool for predicting the students' performance at the final examinations. For this purpose, we have adopted the following methodology that consists of three stages.

The first stage of the proposed methodology concerns the data collection and data preparation for this research followed by the model construction stage. In this stage, we evaluate the classification performance of the most popular and frequently used algorithm for each described machine learning technique by conducting a series of tests. In the final stage, the classifier with the best accuracy is incorporated in a user-friendly software tool for the prediction of students' performance in order to make easier for an educator to identify the weak students and propose supportive actions.

Dataset

The data used in our study concern the students' performance in Mathematics of the first year of Lyceum that is students of ages 14-15 years. The data have been collected by the private Lyceum "Avgoulea-Linardatou" during the years 2007-2010 and consists of 279 different patterns. The attributes concern information about the students' performance such as oral grades, tests grades and final examination grades. Table 1 presents the set of attributes that are divided in two main represent the set of attributes concerning the students' performance on the first and second semester respectively.

Furthermore, the students were classified using a four-level classification, according to the classification scheme used in students' performance evaluation in the Greek schools, namely

- "Fail" stands for student's performance between 0 and 9.
- "Good" stands for student's performance between 10 and 14.
- "Very good" stands for student's performance between 15 and 17.
- "Excellent" stands for student's performance between 18 and 20.

Table 1. List of attributes used in our study

Student's attributes of the $1^{st}/2^{nd}$ Semester	Range values
The oral grade of the $1^{st}/2^{nd}$ semester	[0,20]
The grade of the 1st test of the $1^{st}/2^{nd}$ semester	[0,20]
The grade of the 2nd test of the $1^{st}/2^{nd}$ semester	[0,20]
The grade of the final examination of the $1^{st}/2^{nd}$ semester	[0,20]
The final grade of the $1^{st}/2^{nd}$ semester	[0,20]

Figure 1 presents the class distribution which depicts the number of students who are classified as "Fail" (53 instances), "Good" (76 instances), "Very good" (85 instances) and "Excellent" (65 instances).

Since it is of great importance for an educator to recognize weak students in the middle of the academic period, two datasets have been created based on the attributes presented in Table 1 and on the class distribution.

- $DATA_A$: It contains the attributes which concern the student's performance of the 1st semester.
- $DATA_{AB}$: It contains the attributes which concern the student's performance of the 1st and 2nd semesters.

Notice that, each dataset in our study is used to create an independent classifier which recognizes weak students.

Figure 1. Class distribution

Evaluation/Experimental results

Next, we conduct a series of tests in order to establish which learning algorithm predicts the class (“Fail”, ”Good”, “Very good”, “Excellent”) in which a student belongs based on its grades on both academic semesters. Thus, we have selected the most popular and frequently used algorithm for each described machine learning technique.

The most commonly used Naive Bayes (NB) algorithm was the representative of the Bayesian networks (Domingos & Pazzani, 1997). It is a simple learning algorithm that captures the assumption that every attribute is independent from the rest of the attributes, given the state of the class attribute. The back propagation algorithm (BP) with momentum (Rumelhart et al., 1986) was representative of the ANNs which has been established as a well-known learning algorithm for building a neural network (Lerner et al., 1999). The RIPPER algorithm (Cohen, 1995) was the representative of the rule-learning techniques because it is one of the most usually used methods for producing classification rules. RIPPER forms rules through a process of repeated growing and pruning while the grow heuristic used in RIPPER is the information gain function. We also used the 3NN algorithm, with Euclidean distance as distance metric as instance-based learner (Aha, 1997). From the decision trees, C4.5 algorithm (Quinlan, 1993) was the representative in our study. C4.5 algorithm uses a statistical property known as information gain at each level in the partitioning process in order to determine which attribute best divides the training examples. Finally, from the SVMs we have selected the Sequential Minimal Optimization (SMO) algorithm in our study since it is one of the fastest methods to train SVMs (Platt, 1999). For evaluating classification accuracy we have used the standard procedure called 10-fold cross-validation (Kohavi, 1995) and all algorithms have been implemented in WEKA toolbox (Hall et al., 2009). In order to minimize the effect of any expert bias by not attempting to tune any of the algorithms to the specific datasets we have utilized the default values of all learning parameters.

Table 2 summarizes the performance of each classifier, measured by the percentage of patterns that were classified correctly in the presented datasets. Clearly, no single algorithm can perform well and uniformly outperform the other algorithms. More specifically, 3NN presents the best performance as regards dataset $DATA_A$, while BP reports the highest percentage of correctly classified instances relative to dataset $DATA_{AB}$.

Since our main goal is to generate more precise and accurate system results, we combine the predictions of the individual algorithms on the presented datasets utilizing voting, stacking and grading methodology. The methods in the first column in Table 3 have the following meaning:

Table 2. Classifiers’ accuracy for each dataset

	Datasets	
Classifiers	$DATA_A$ (%)	$DATA_{AB}$ (%)
NB	51.6	59.5
BP	58.1	70.3
RIPPER	57.0	67.0
3NN	59.9	67.7
C4.5	56.6	68.1
SMO	57.0	64.2

- BestCV stands for the methodology of selecting the best classifier (Witten, Frank & Hall, 2005).
- Voting stands for simple voting methodology combining the prediction of the individual algorithms presented in Table 2.
- Stacking stands for stacking methodology using the same base classifiers as Voting and MLR as meta-level classifier (Ting & Witten, 1999).
- Grading stands for grading methodology utilizing the same base classifiers as Voting and the instance base classifier 10NN as the meta-level classifier (Seewald & Furnkranz, 2001).
- Voting* stands for simple voting methodology using RIPPER, 3NN, BP and SMO as base classifiers.
- Stacking* stands for stacking methodology using the same base classifiers as Voting* and MLR as meta-level classifier (Ting & Witten, 1999).
- Grading* stands for grading methodology utilizing as base classifiers as Voting* and the instance base classifier 10NN as the meta-level classifier (Seewald & Furnkranz, 2001).

The interpretation of Table 3 indicates that the Voting methodology is more accurate than the other ensembles, exhibiting the best performance with Voting* significantly outperforming all algorithms regarding both datasets.

Decision support tool

In this section, we present a prototype version of our software support tool for predicting the students' performance at the final examinations (Figure 2). The tool has been developed in JAVA and it is based on the WEKA Machine Learning Toolkit (Hall et al., 2009); thus it can be deployed in any platform with the minimum requirement of having installed a Java virtual machine. Notice that the classifiers incorporated in this software tool are based on the Voting* methodology which presented the best generalization performance.

Table 3. Ensembles' accuracy for each dataset

	Datasets	
Classifiers	$DATA_A$ (%)	$DATA_{AB}$ (%)
BestCV	59.9	70.3
Voting	58.1	86.0
Stacking	56.6	68.5
Grading	57.7	71.7
Voting*	60.6	90.3
Stacking*	57.7	71.7

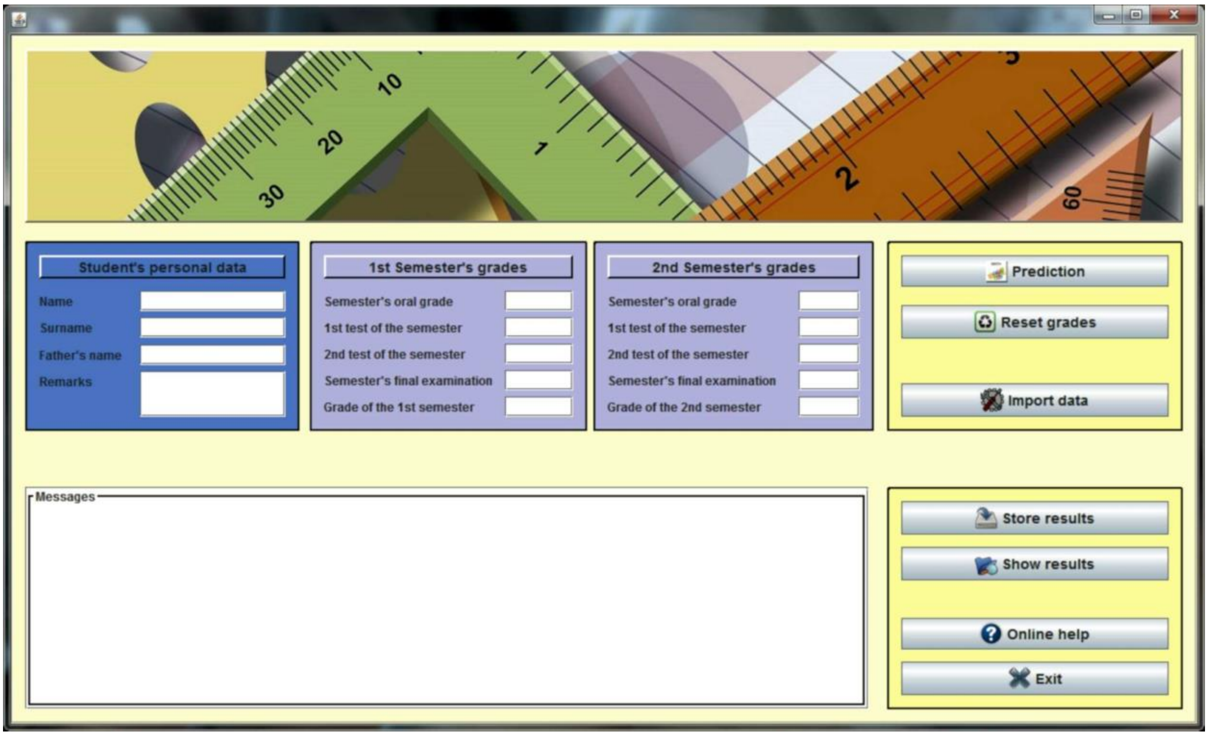

Figure 2. The prediction Tool

The main features of our software tool are:

- Student personal data: this module is used to import the personal information of the student such as name, surname and father's name.
- 1st Semester's grades: this module is used to import the student's grades in the first semester.
- 2nd Semester's grades: this module is used to import the student's grades in the second semester.
- Messages: this module is used to print the messages, warnings and outputs of the tool.

Next, we demonstrate a use case in order to illustrate the functionality of our decision support tool. Firstly, the user/educator by clicking on the button "Import data" can load his/her data collected from his/her own past courses or use our data embedded in the tool (Figure 3). In case the user selects to use his/her own data, the tool expects the data in XLSX (Microsoft Office Excel 2007 XML) file format. If the first row of the XLSX file is used for the names of the attributes then the tool automatically ignores it. Subsequently, the tool is constructing the classifiers for predicting the student's performance in the final examinations. The first classifier is trained using the students' data from the 1st semester, while the second classifier is trained using the data from both semesters.

After the classifiers are developed, the user can import the new students' grades of both academic semesters in the corresponding fields. Next, by clicking on the button ''Prediction" the educator can predict a student's performance at the final examinations. It is worth noticing that since it is of great importance for an educator to early identify weak students, the tool has the feature of predicting a student's performance utilizing only the grades of the 1st semester. Figure 4 presents an example in which the model predicts that the student is classified as "Good" based on the student's grades of the 1st semester.

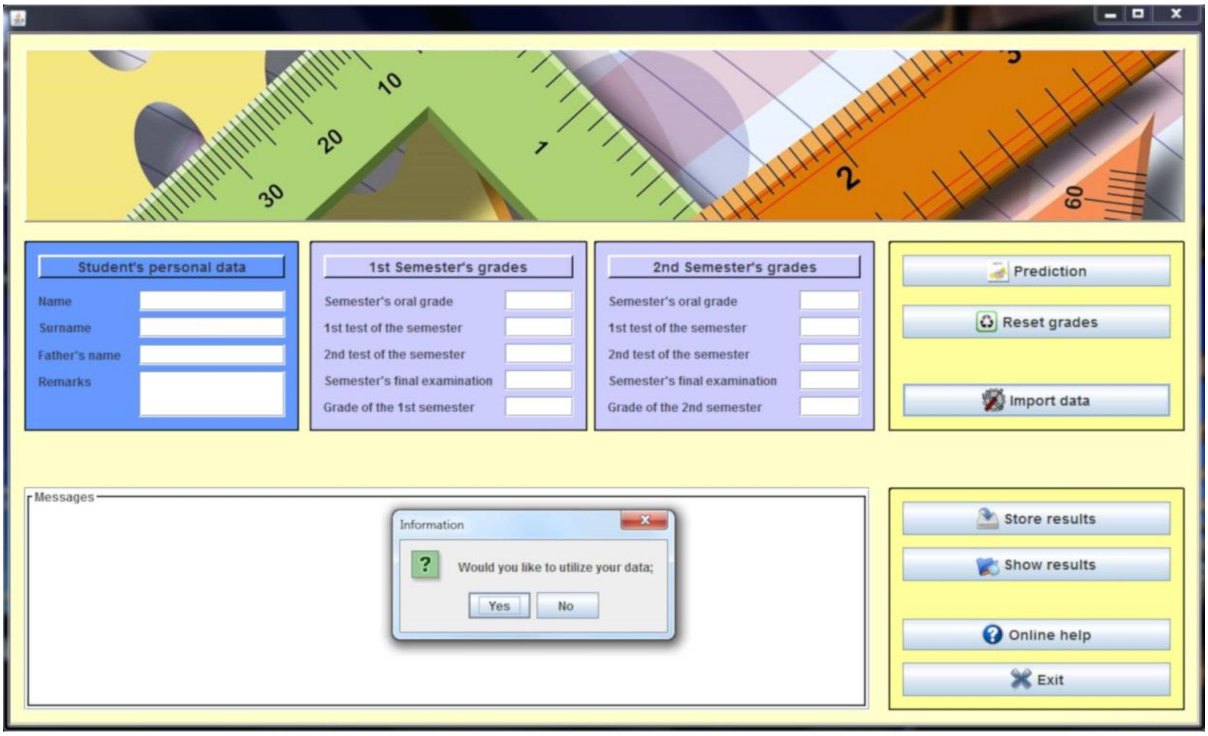

Figure 3. Selecting training data and importing them in the prediction tool

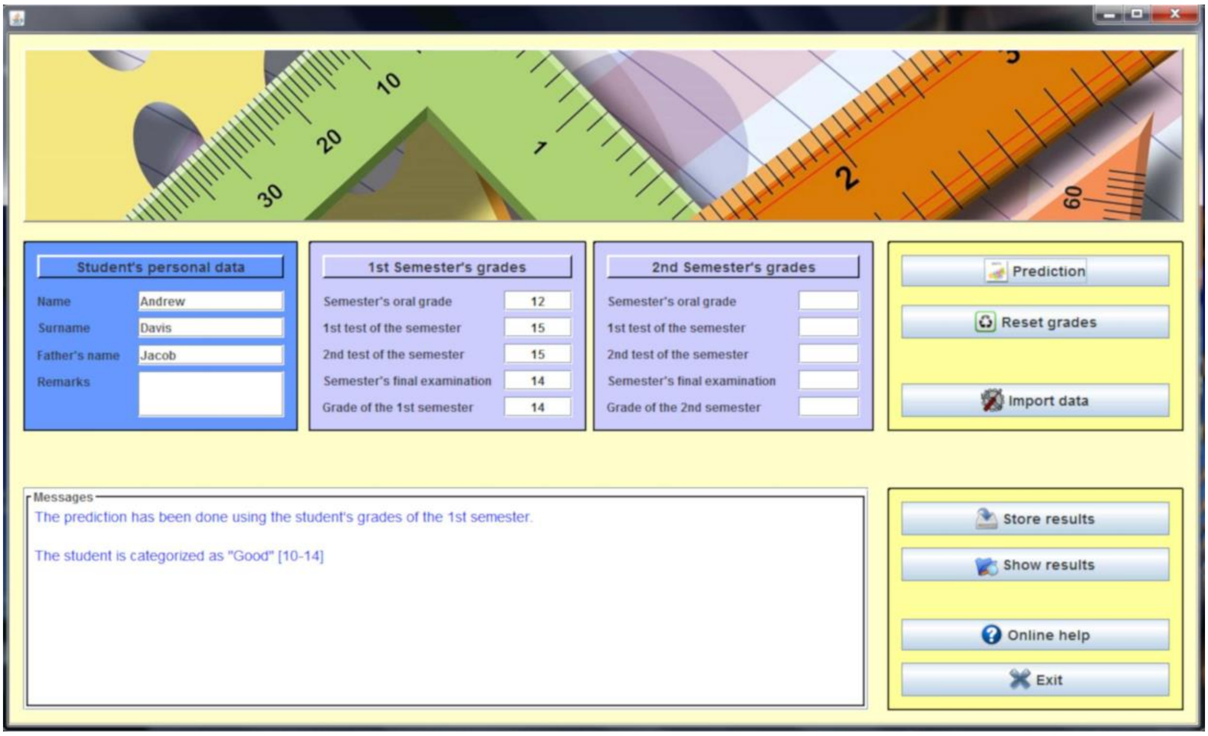

Figure 4. Tool's prediction about the performance of a new student at the final examinations utilizing the grades of the 1st Semester

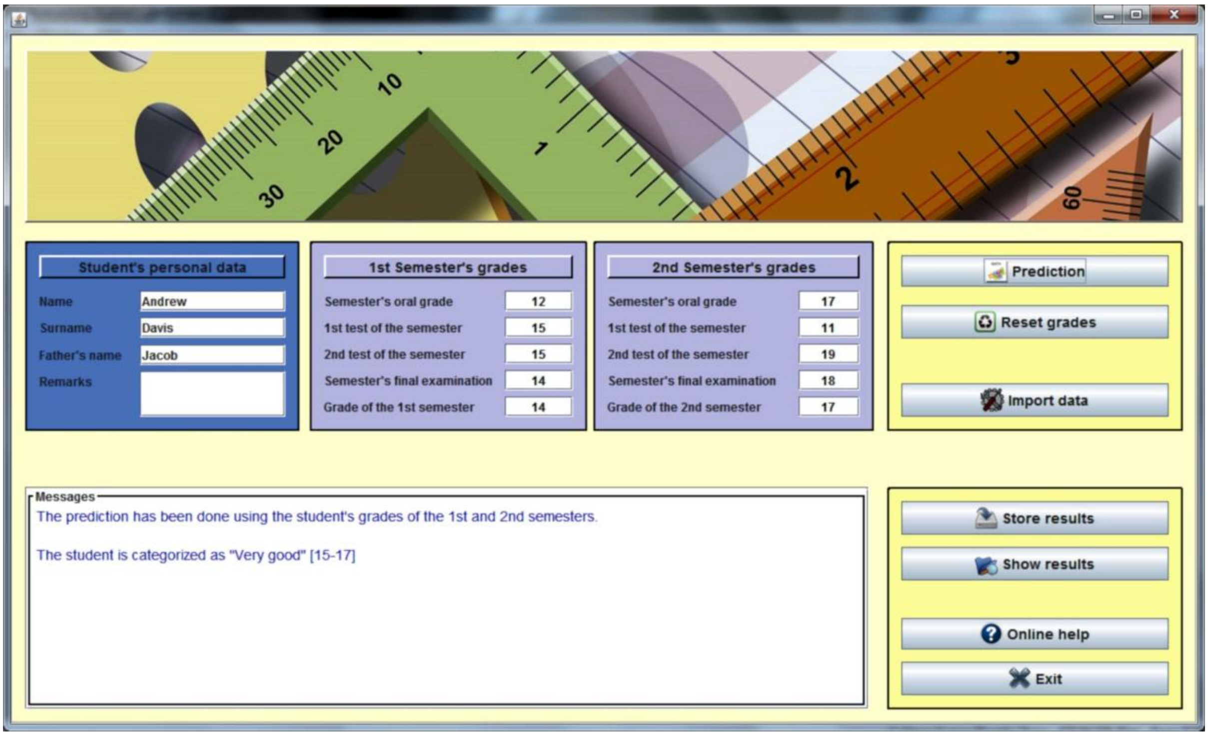

Figure 5. Tool's prediction about the performance of a new student at the final examinations utilizing the grades of the 1st and 2nd Semesters

Similarly, the educator can have the tool's prediction with higher accuracy by also importing the grades of the 2nd semester. In the example presented in Figure 5, the model predicts that the student is classified as "Very good" based on the student's grades of both academic semesters. Additionally, the tool provides the ability to store all the predictions by simply clicking on the button "Store results" in order to activate this feature. In this case, the predictions for each student are stored in a XLSX file and a TXT file. Moreover, the user has also the ability to see all previous results by clicking the button "Show results" (Figure 6).

Conclusion and future research

Prediction, using machine learning and data mining computational techniques, is a significant tool and represents a first step and a helping hand in intervention from the educators to early recognize those students who are likely to exhibit poor performance. In this work, we developed a user-friendly decision support tool for predicting the student's performance, together with a case study concerning the final examinations in Mathematics of the first year of Lyceum. Our proposed tool is based on a hybrid predicting system incorporating a number of possible machine learning methods and achieves better performance than any examined single learning algorithm. Furthermore, significant advantages of the presented tool are that it has a simple user interface and it can be deployed in any platform under any operating system while this is not the case with any other similar attempt (Kotsiantis, 2012; Livieris et al. 2012; Pandey & Taruna, 2014). We have illustrated the main features of our software tool and we have also presented a case study to illustrate its functionalities and the experiment set up processes. Our preliminary results revealed that we can early gain insights about student progress and recommend possible actions such as further study or additional learning activities, resources and learning tasks.
Furthermore, it is worth mentioning that the used attributes in the support tool are not a conclusive list. An extension can introduce new attributes that were not in the current

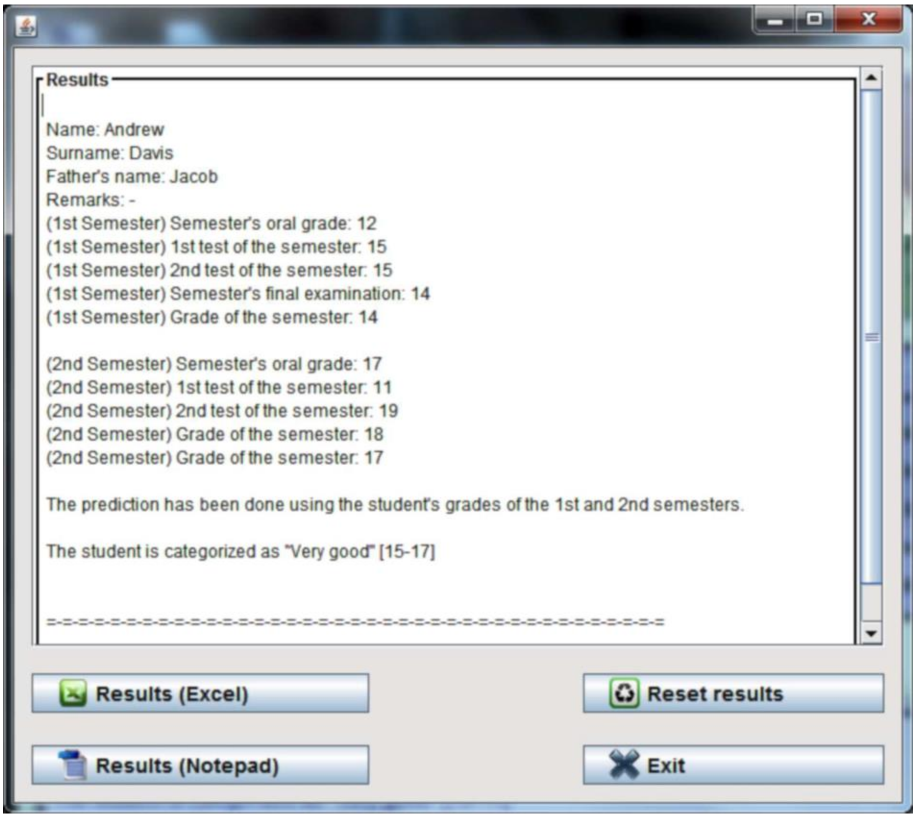

Figure 6. Stored predictions about students' performance at the final examination

database, but are collectable by tutors and may potentially contribute to the prediction of student's performance i.e. more tests, homeworks, projects.

Currently, our prediction tool is still under development and given that this is a pilot study, our evaluator sample (teachers and educators) is rather small. Hence, in our plans is to do a systematic and extensive evaluation of the tool by several groups of external teachers in order to evaluate its usability. Moreover, another direction for a future research would be to collect data from all three years of Lyceum and apply our methodology for predicting the students' performance at PanHellenic (national) level examinations for admission to Universities.

Acknowledgments

The authors are grateful to the private Lyceum "Avgoulea-Linardatou" for the collection of the data used in our study, for the evaluation of the tool and for their kind and valuable comments, which essentially improved our work.

References

Aha, D. (1997). *Lazy Learning*. Dordrecht: Kluwer Academic Publishers.

Aha, D., Kibler, D., & Albert, M. (1991). Instance-based learning algorithms. *Machine Learning*, 6(1), 37–66.

Anju, R., & Robin, P. (2013). Survey on decision tree classification algorithms for the evaluation of student performance. *International Journal of Computers & Technology*, 4(2), 244–247.

Baker, R. S., & Inventado, P. S. (2014). Educational Data Mining and Learning Analytics. In J. A. Larusson & B. White (Eds.), *Learning Analytics: From Research to Practice*. (pp. 61–75). Heidelberg: Springer-Verlag.

Baker, R. &Yacef, K. (2009). The state of educational data mining in 2009: A review future visions. *Journal of Educational Data Mining*, 1(1), 3–17.

Baradwaj, B. &Pal, S. (2011). Mining educational data to analyse student performance. *International Journal of Advanced Computer Science and Applications*, 2(6), 63–69.

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.

Chen, J. & Do, Q. (2014). Training neural networks to predict student academic performance: A comparison of cuckoo search and gravitational search algorithms. *International Journal of Computational Intelligence and Applications*, 13(1), doi: 10.1142/S1469026814500059.

Cohen, W. W. (1995). Fast effective rule induction. In A. Prieditis & S. Russell (Eds.), *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 115–123). CA: Morgan Kaufmann.

Cortez, P. & Silva, A. (2008). Using data mining to predict secondary school student performance. In A. Brito & J. Teixeira (Eds.), *Proceedings of 5th Annual Future Business Technology Conference* (pp. 5-12). Universidade do Minho: EUROSIS.

Dietterich, T. (2001). Ensemble methods in machine learning. In J. Kittler, & F. Roli (Eds.), *Multiple Classifier Systems* (pp. 1–15). Heidelberg: Springer-Verlag.

Domingos, P. & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29, 103–130.

Furnkranz, J. (1997). Pruning algorithms for rule learning. *Machine Learning*, 27, 139–171.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. (2009). The WEKA data mining software: An update. *SIGKDD Explorations Newsletters*, 11, 10–18.

Russell, M., Bebell, D., O'Dwyer, L., & O'Connor, K. (2003). Examining teacher technology use. Implications for preservice and inservice teacher preparation. *Journal of Teacher Education*, 54(4), 297-310.

Haykin, S. (1994). *Neural Networks: A comprehensive foundation*. New York: Macmillan College Publishing Co.

Jensen, F. (1996). *An Introduction to Bayesian Networks*. Heidelberg: Springer-Verlag.

Kabra, R. & Bichkar, R. (2011). Performance prediction of engineering students using decision trees. *International Journal of Computer Applications*, 36, 8–12.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IEEE International Joint Conference on Artificial Intelligence* (pp. 1137–1143). CA: Morgan Kaufmann.

Kohavi, R. & Quinlan, J. (1999). Decision tree discovery. In *Handbook of data mining and knowledge discovery* (pp. 267–276). USA: University Press.

Kotsiantis, S. (2007). Combining bagging and additive regression. *International Journal of Mathematics Science*, 1, 61–67.

Kotsiantis, S. (2012). Use of machine learning techniques for educational proposes: a decision support system for forecasting students' grades. *Artificial Intelligence Review*, 37, 331–344.

Kotsiantis, S., Pierrakeas, C., & Pintelas, P. (2003). Preventing student dropout in distance learning using machine learning techniques. In V. Palade, R. J. Howlett, & L. C. Jain (Eds.), *Knowledge-Based Intelligent Information and Engineering Systems* (pp. 267–274). Heidelberg: Springer-Verlag.

Kotsiantis, S., Pierrakeas, C., & Pintelas, P. (2004). Predicting students' performance in distance learning using machine learning techniques. *Applied Artificial Intelligence*, 18(5), 411–426.

Kuncheva, L. (2014). *Combining Pattern Classifiers: Methods and Algorithms*. NJ: John Wiley & Sons, Inc.

Lerner, B., Guterman, H., Aladjem, M., & Dinstein, I. (1999). A comparative study of neural network based feature extraction paradigms. *Pattern Recognition Letters*, 20(1), 7–14.

Livieris, I., Drakopoulou, K., & Pintelas, P. (2012). Predicting students' performance using artificial neural networks. In Ch. Karagiannidis, P. Politis, El. Karasavidis (Eds.), *8th PanHellenic Conference with International Participation Information and Communication Technologies in Education* (pp. 321-328). Ioannina: ETPE.

Livieris, I., Mikropoulos, T., & Pintelas, P. (2016). A decision support system for predicting student's performance. *Technical Report, TR16-01*, University of Patras.

Merz, C. (1997). Combining classifiers using correspondence analysis. In M. I. Jordan, M. J. Kearns, & S. A. Solla (Eds.), *NIPS '97 Proceedings of the 1997 conference on Advances in neural information processing systems 10* (pp. 591–597). MA: MIT Press.

Merz, C. (1999). Using correspondence analysis to combine classifiers. *Machine Learning*, 36, 33-58.

Mitchell, T. (1997). *Machine Learning*. USA: McGraw Hill.

Mohamad, S., & Tasir, Z. (2013). Educational data mining: A review. *Procedia-Social and Behavioral Sciences*, 97, 320–324.

Murthy, K. (1998). Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2, 345–389.

Oladokun, V., Adebanjo, A., & Charles-Owaba, O. (2008). Predicting students' academic performance using artificial neural network: A case study of an engineering course. *The Pacific Journal of Science and Technology*, 9(1), 72–79.

Pandey, M. & Taruna, S. (2014). A multi-level classification model pertaining to the student's academic performance prediction. *International Journal of Advances in Engineering & Technology*, 7(4), 13–29.

Platt, J. (1999). Using analytic QP and sparseness to speed training of support vector machines. In M. S. Kearns, S. A. Solla, & D. A. Cohn, (Eds.), *Proceedings of the 1998 conference on Advances in neural information processing systems II* (pp. 557–563). MA: MIT Press.
Quinlan, J. (1993). *C4.5: Programs for machine learning*. CA: Morgan Kaufmann.
Rokach, L. (2010). *Pattern Classification Using Ensemble Methods*. London: World Scientific Publishing Company.
Romero, C. & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33, 135–146.
Romero, C. & Ventura, S. (2010). Educational data mining: A review of the state of the art. *IEEE on Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 40(6), 601–618.
Romero, C., Ventura, S., Pechenizkiy, S., & Baker, M. (2010). *Handbook of Educational Data Mining*. London: Chapman & Hall.
Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning internal representations by error propagation. In D. Rumelhart, & J. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (pp. 318–362). MA: MIT Press.
Seewald, A. & Furnkranz, J. (2001). An evaluation of grading classifiers. In F. Hoffmann, D. J. Hand, N. Adams, D. Fisher, & G. Guimaraes (Eds.), *Advances in Intelligent Data Analysis: Proceedings of the 4th International Conference* (pp. 221–232). Heidelberg: Springer-Verlag.
Ting, K. & Witten, I. (1999). Issues in stacked generalization. *Artificial Intelligence Research*, 10, 271–289.
Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Heidelberg: Springer-Verlag.
Witten, I., Frank, E., Hall, M. A. (2005). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. CA: Morgan Kaufmann.
Wolpert, D. (1992). Stacked generalization. *Neural Networks*, 5(2), 241–260.

Appendix

The tool is available in the web page http://www.math.upatras.gr/~livieris/EducationalTool/Tool.zip. Notice that Java Virtual Machine (JVM) 1.2 or newer is needed for the execution of the program.

To cite this article: Livieris, I. E., Mikropoulos, T. A., & Pintelas, P. (2016). A decision support system for predicting students' performance. *Themes in Science and Technology Education*, 9(1), 43-57.

URL: http://earthlab.uoi.gr/theste