Improving Academic Performance Prediction by Dealing with Class Imbalance

Nguyen Thai-Nghe, Andre Busche, and Lars Schmidt-Thieme
Information Systems and Machine Learning Lab
University of Hildesheim
Samelsonplatz 1, 31141 Hildesheim, Germany
{*nguyen, busche, schmidt-thieme*}*@ismll.de*

***Abstract*—This paper introduces and compares some techniques used to predict the student performance at the university. Recently, researchers have focused on applying machine learning in higher education to support both the students and the instructors getting better in their performances. Some previous papers have introduced this problem but the prediction results were unsatisfactory because of the class imbalance problem, which causes the degradation of the classifiers. The purpose of this paper is to tackle the class imbalance for improving the prediction/classification results by over-sampling techniques as well as using cost-sensitive learning (CSL). The paper shows that the results have been improved when comparing with only using baseline classifiers such as Decision Tree (DT), Bayesian Networks (BN), and Support Vector Machines (SVM) to the original data sets.**

***Keywords*-Academic performance; Prediction; Class imbalance; Cost-sensitive.**

I. Introduction

Machine learning has been applied in plenty of areas. For examples, it can be used to predict/classify customer behaviors in marketing or sales, to detect fraudulent or default in banking, to diagnose diseases in medicine, and recently, to a new area which is education. Moreover, the universities desire to improve their educational quality, hence, how to apply machine learning techniques in higher education to help the universities, instructors, and the students getting better in their performances become more and more attractive to both university managers and researchers.

In the first conference of educational data mining 2008, [1] compared different data mining methods and techniques to classify students based on their Moodle usage data and the final marks obtained in their respective courses; [2] proposed a model with different types of education-related questions and the data-mining techniques appropriate for them. For examples, predicting student performance, clustering similar students, and associating types of students with appropriate courses; [3], [4], and [5] used BN, DT, and other common techniques to predict the student results; another study [6] was done by us recently to predict the student performance at two real case studies: Can Tho University, Vietnam (CTU)[1], and Asian Institute of Technology, Thailand (AIT)[2].

Researchers have spent a lot of time in the task of predicting student performances while archiving only fair results. One major problem is that the number of ”pass students” is much higher than the number of ”fail students”. This skew distribution is the main reason causing the degradation of classifiers.

The purposes of this study are both to improve the results of classification techniques in predicting the student performance of two real world case-studies (CTU, AIT) by dealing the class imbalance problem, and to compare the results of some common classification techniques before and after approaching with class imbalance.

The rest of the paper is organized as follows. Section 2 introduces some common techniques used to tackle with the class imbalance problem; section 3 represents our methods; section 4 explores the datasets; section 5 shows the results; finally, section 6 and 7 represent discussion and conclusion respectively.

II. Dealing with class imbalance problem

To deal with imbalanced datasets, researchers used to focus on data level and classifier (algorithm) level. At data level, the common task is the modification of class distribution using over-sampling or under-sampling techniques. At classifier level, some common techniques were introduced such as manipulating classifiers internally, one-class learning, ensemble methods, and CSL [7]. We will briefly review some techniques used in this study.

A. Modifying class distribution

To modify the class distribution, over-sampling and under-sampling are usually used. Under-sampling discards a lot of useful information and usually appropriates for large datasets. The most common over-sampling method is random over-sampling, which randomly duplicates the examples in the datasets. Another over-sampling method was introduced by [8] and called SMOTE (Synthetic Minority Over-sampling Technique). SMOTE generates new artificial minority examples by interpolating between the existing minority examples rather than simply duplicating the original examples. This method, at first, finds k nearest neighbors of each minority example (according to authors, k=5); then it selects a random nearest neighbor; finally, the new synthetic

[1] http://www.ctu.edu.vn
[2] http://www.ait.ac.th

examples are generated along the line segment joining a minority class sample and its nearest neighbor.

B. Cost-sensitive learning

Most of classifiers assume that the misclassification costs (false negative and false positive cost) are the same. In some real-world applications, this assumption may not be true. For examples, the cost of mailing to non-buyers is less expensive than the cost of non-mailing to the buyers [9]; or the cost of predicting non-terrorist to terrorist is much cheaper than the cost of misclassifying an actual terrorist who carries a bomb to a flight. Cost is not necessarily monetary, for examples, it can be a waste of time or even the severity of an illness [10]. In our studies, the cost of misclassifying the actual ”fail students” to ”pass students” (so we can not help them, consequently, they will be expelled from the university) is much costlier than the false alarm.

When learning with two-class problems, we assume that the positive class $(+)$ represents for the minority examples, and the negative class $(-)$ represents for the majority examples. Let $C(i,j)$ be the cost of predicting an example belonging to class i when in fact it belongs to class j, then the confusion matrix and the cost matrix are described as in tables I and II respectively.

Table I
CONFUSION MATRIX

		Predict classes	
		Positive	Negative
Actual classes	Positive	TP	FN
	Negative	FP	TN

Table II
COST MATRIX

		Predict classes	
		Positive	Negative
Actual classes	Positive	$C(+,+)$	$C(-,+)$
	Negative	$C(+,-)$	$C(-,-)$

Given this cost matrix, an example x can be classified into class i with the minimum expected cost (*conditional risk*) by using Bayes risk in equation (1)

$$\mathcal{H}(x) = \arg\min_i \left(\sum_j^N P(j|x)C(i,j) \right) \tag{1}$$

where N is number of classes and $P(j|x)$ is the posterior probability of classifying an example x into class j.

Another term is the cost ratio, which is the proportion of false negative and false positive determined by $C(-,+)/C(+,-)$. The purpose of cost-sensitive learning is to build the model with minimum misclassification cost as described in equation (2)

$$Totalcost = C(-,+) \times FN + C(+,-) \times FP \tag{2}$$

C. Evaluation metrics

In the case of imbalanced datasets, accuracy metric becomes useless. For example, suppose the dataset has 990 negative examples and only 10 positive examples (this minority is usually very important). Most of the classifiers designed to maximize the accuracy, so in this case, they will classify all examples belong to the majority class to get the maximum of 99% accuracy. This result has no meaning because all the positive examples are classified incorrectly. To evaluate the model in the case of class imbalance, researchers usually use F-measure and the AUC, which are related to some other metrics described in the following.

False negative rate (FNR) is the proportion of positive examples misclassified as belonging to the negative class, $FNR = \frac{FN}{TP+FN}$

True negative rate (TNR) is the proportion of negative examples correctly classified as belonging to the negative class, $TNR = \frac{TN}{FP+TN)}$

False Positive Rate (FPR) is the proportion of negative examples misclassified as belonging to the positive class, $FPR = \frac{FP}{TN+FP}$

True Positive Rate (TPR) (or Recall R) is the proportion of positive examples correctly classified as belonging to the positive class, determined by:

$$TPR = R = \frac{TP}{TP+FN} \tag{3}$$

Precision (P) is the positive predictive value determined by:

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

F-Measure is an evaluation metric which considers both precision and recall of the testing results (β is used to set equal to 1)

$$F - Measure = \frac{(1+\beta^2) \times P \times R}{(\beta^2 \times P + R)} \tag{5}$$

Another important metric is the area under the ROC curve (AUC) [11], in which ROC (Receiver Operating Characteristic) curve is a graphical approach for displaying the tradeoff between TPR (y-axis) and FPR (x-axis) of a classifier. We will use these metrics to evaluate our models in section 5.

III. METHODS

The first step in this study is to collect the real datasets from the relational databases. Based on experts and feature selections, we have selected the most important attributes for the prediction tasks. The second step is to apply techniques to tackle the class imbalance based on three methods:

- Method 1: Modifying the class distribution by using SMOTE.
- Method 2: Applying CSL to minimize the total misclassification cost.
- Method 3: Combining SMOTE and CSL

The third step is to evaluate and compare the results of these methods by experimenting with 3 classifiers on both the original and the re-balanced datasets. We use an evaluation protocol visualized as in the figure 1. This evaluation works as 10-folds cross-validation. At first, the dataset was separated; next, the enrichment data was generated on the train set; then the classifiers were built; finally, the evaluations were applied to the original test data. The final results were collected from average of ten run-times.

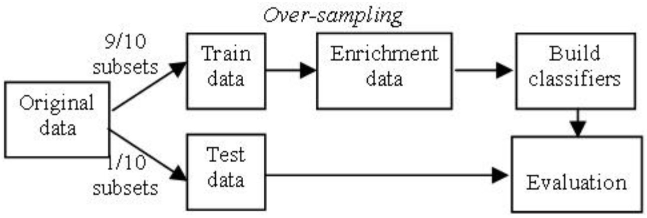

Figure 1. Evaluation protocol

The final step in this research is to build the intelligent system based on the best model we have evaluated. This system is deployed on the local web to help not only the students in predicting their result to have clever study plans but also the instructors who give them advices or more tutorials.

IV. Datasets

The first dataset was got from Can Tho University, Vietnam (CTU), which has 20492 examples and 14 attributes. The second dataset was collected from the Asian Institute of Technology, Thailand (AIT), which has 936 examples and 14 attributes. We use these two datasets for predicting the student results (”pass”, ”fail”).

Furthermore, to observe the results on the less imbalance data, two other datasets were also collected. The third dataset was taken from UCI repository[3] named Teaching Assistant Evaluation (TAE). This data consists of evaluation of teaching performance for 151 teaching assistant assignments at University of Wisconsin-Madison. The scores were divided into 2 classes (”low”, ”med-high”). The fourth dataset was collected from the Journal Statistics of Education Data Archive[4] named ”U.S. News College” (USNC) having 1204 colleges, 33 attributes. This data is used to classify the graduation rate. We artificially assigned binary target labels for USNC by dividing the students at 85% graduation rate to get the imbalanced ratio between two real datasets AIT/CTU and TAE, as in table III

V. Experimential results

The experiments and the application system in this study were developed based on the WEKA library.[5]

Table III
Datasets

Dataset	#Examples	#Attributes	#minor	#major/ #minor
CTU	20492	14	1565	13.09
AIT	936	14	75	12.48
TAE	151	6	49	3.08
USNC	1024	33	132	7.76

A. Hyperparameters search

We have applied hyperparameter searches to look for the best parameters on all classifiers. To observe how the classifiers are influenced by the each dataset, we have also searched for the best parameter (percentage) of oversampling techniques. Each dataset has its own structure, so the percentage of undersampling and oversampling are also different. This percentage is treated as hyperparameter. For oversampling, we search on the percentage from 50, 100, 200,.. to a balanced distribution. There are some notations such as CTU-SM150 represents for the dataset with SMOTE at 150%, and CTU-O means that the Original CTU dataset was used.

Table IV shows the best hyperparameters for DT. The ”unpruned” indicates whether pruning is performed; ”use-Laplace” indicates whether counts at leaves are smoothed based on Laplace [12].

Table IV
Hyperparameters for DT

Dataset	Un pruned	use Laplace	Binary split	#Instances per leaf
CTU-O	True	True	True	5
CTU-SM150	False	True	True	10
AIT-O	True	True	True	2
AIT-SM200	True	True	False	2
TAE-O	True	True	False	2
TAE-SM50	True	False	True	2
USNC-O	True	True	False	7
USNC-SM100	True	True	True	3

Table V describes the hyperparameters for BN. Alpha is used for estimating the probability tables and can be interpreted as the initial count on each value [13]; MaxNoOfParents (M) sets the maximum number of parents a node in the BN can have. (M = 1 for Naive Bayes classifier; M = 2 for Tree Augmented Bayes Network [14]; $M > 2$ for Bayes Net Augmented Bayes Network [15]; The framework K2 developed by [16] for induction of BN from data is used.

Given a dataset $\mathcal{D}$ consisting of n examples (x_i, y_i), where $x_i \in \mathcal{X}$ input features and y_i the target class, $y_i \in \mathcal{Y} = \{-1, +1\}$, SVM predicts a new example x by using the function:

$$f(x) = sign\left(\sum_{i=1}^{n} \alpha_i y_i k(\mathbf{x}, \mathbf{x_i}) + b\right) \qquad (6)$$

[3] http://archive.ics.uci.edu/ml

[4] http://www.amstat.org/publications/jse/index.html

[5] http://www.cs. waikato.ac.nz/ml/weka/

Table V
HYPERPARAMETERS FOR BN

Dataset	MaxNoOfParents	Search Algorithm	Alpha
CTU-O	2	Local.K2	0.3
CTU-SM150	3	Local.K2	0.3
AIT-O	2	Local.K2	0.4
AIT-SM200	2	Local.K2	0.1
TAE-O	2	Local.K2	0.2
TAE-SM50	2	Local.K2	0.1
USNC-O	1	Local.K2	0.1
USNC-SM200	1	Local.K2	0.7

where $k(\mathbf{x}, \mathbf{x_i})$ is a kernel function, b is the bias, and α_i is determined by solving the Lagrangian optimization problem, $L_p =$

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i^n \xi_i - \sum_i^n \alpha_i\{y_i(x_i.w+b) - 1 + \xi_i\} - \sum_i^n \mu_i\xi_i \tag{7}$$

where ξ_i is a slack variable, μ_i is Lagrange multiplier, and C is user-specified parameter representing the penalty of misclassifying the training instances and it can be chosen based on calibration.

For non-linear problems, the kernel k is used to maximum margin hyperplanes. Two commonly used kernel functions are the polynomial kernel

$$k(\mathbf{x}, \mathbf{x_i}) = (\gamma \mathbf{x} \cdot \mathbf{x_i} + r)^p \tag{8}$$

and the radial basis function kernel

$$k(\mathbf{x}, \mathbf{x_i}) = e^{-\gamma\|\mathbf{x}-\mathbf{x_i}\|^2} \tag{9}$$

We have searched for the best hyperparameters C, exponent p, and γ in equations (7), (8), and (9) respectively using the method proposed by [17]. At first, a "raw search" on the powers of two $(2^{-4} \dots 2^8)$ for C values was used to identify a "good region", then a "smooth search" around that region was conducted. Table VI represents these results, in which bLogistic indicates whether to fit logistic models to the outputs.

Table VI
HYPERPARAMETERS FOR SVM

Dataset	Kernel	p/γ	C	bLogistic
CTU-O	Poly	1	2.5	Yes
CTU-SM200	Poly	1	2.5	Yes
AIT-O	Poly	1	1.6	Yes
AIT-SM200	Poly	1	2.0	Yes
TAE-O	Poly	3	3.0	Yes
TAE-SM80	Poly	1	2.0	Yes
USNC-O	RBF	0.01	2.0	Yes
USNC-SM50	RBF	0.01	1.0	Yes

When learning with cost-sensitive, this study also searches for the cost ratio between false negative and false positive. We will investigate more clearly in next section.

B. *Evaluation and comparison*

We have experimented with three classifiers on four datasets. The AUC and F-Measure results are shown in table VII. In each dataset, this study shows that the classification results after dealing the class imbalance problem are normally better than learning on original data.

Another important method to deal with class imbalance is CSL, in which the classifiers will not consider the FN and FP equally. In our application, FN is the misclassification cost from the actual "fail student" to "pass student" and FP is the misclassification from the "pass" to the "fail" one. What is the cost in this case? Suppose that a second year student fails in this year, if he will fail again in next year then he will be expelled from the university, so the cost of both student and the university can be estimated as FN-Cost = (tuition fees + living fees + insurance fees + other fees) * 2-years-have-passed + (tuition fees will be lost from the university) * 2-years-in-future + (invaluable moral-cost). Conversely, the cost of FP can be approximated as FP-Cost = (the fee of some tutorials).

In fact, the cost of FN is much higher than FP. To examine how the total cost change when we increase the cost ratio, we have experimented on the cost ratio from 2 to 20. Figures 2 (only uses CSL) and 3 (combining SMOTE with CSL) display the total cost produced by DT, BN, SVM, and DT without cost-sensitive (DT-O). The results indicate that DT has the lowest total cost when comparing with the others on original data, and SVM has the lowest total cost on rebalanced data. Total cost is also reduced when learning on over-sampling datasets.

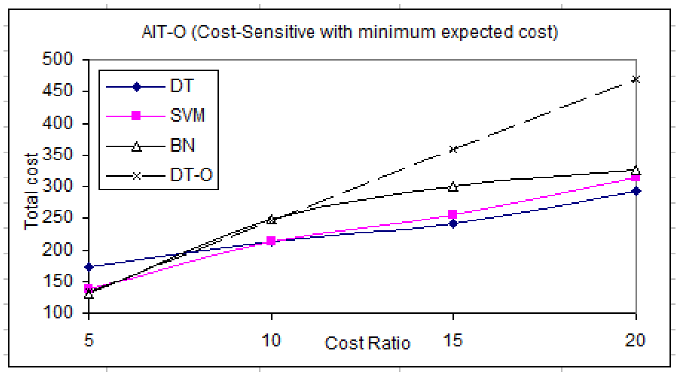

Figure 2. Total cost of AIT-O

To observe clearly how CSL effects on our problems, we have also experimented with three other ensemble methods, which can be used for handling class imbalance problem, AdaBoost [18], Bagging, and MetaCost [10] on the baseline DT. (Since the limitation of space, some other results are not presented here).

Figures 4 and 5 compare the total cost on ensemble methods. AdaBoost, Bagging and original-DT increase the

Table VII
AUC AND F-MEASURE

Algorithm	Dataset	TPR	FPR	Precision	AUC	F-Measure
Decision Tree	AIT-O	0.160	0.063	0.182	0.692 (0.090)	0.170 (0.120)
	AIT-SM200	0.280	0.038	0.389	0.763 (0.060)	0.326 (0.090)
	CTU-O	0.347	0.029	0.507	0.879 (0.010)	0.412 (0.030)
	CTU-SM150	0.466	0.056	0.550	0.882 (0.010)	0.495 (0.020)
	TAE-O	0.375	0.147	0.545	0.660 (0.130)	0.444 (0.080)
	TAE-SM50	0.563	0.382	0.409	0.733 (0.100)	0.474 (0.090)
	USNC-O	0.341	0.022	0.615	0.821 (0.070)	0.455 (0.010)
	USNC-SM100	0.545	0.062	0.522	0.850 (0.030)	0.533 (0.020)
Bayesian Networks	AIT-O	0.080	0.021	0.250	0.655 (0.080)	0.118 (0.050)
	AIT-SM200	0.120	0.031	0.273	0.731 (0.050)	0.167 (0.100)
	CTU-O	0.383	0.025	0.570	0.856 (0.010)	0.458 (0.030)
	CTU-SM50	0.400	0.031	0.535	0.877 (0.010)	0.456 (0.020)
	TAE-O	0.438	0.206	0.438	0.680 (0.130)	0.296 (0.080)
	TAE-SM90	0.625	0.294	0.500	0.688 (0.100)	0.556 (0.090)
	USNC-O	0.514	0.123	0.380	0.848 (0.060)	0.470 (0.020)
	USNC-SM200	0.591	0.115	0.388	0.864 (0.030)	0.531 (0.020)
Support Vector Machines	AIT-O	0.120	0.021	0.250	0.690 (0.016)	0.167 (0.008)
	AIT-SM100	0.200	0.035	0.330	0.739 (0.053)	0.250 (0.092)
	CTU-O	0.385	0.026	0.500	0.813 (0.012)	0.450 (0.020)
	CTU-SM200	0.454	0.038	0.510	0.878 (0.010)	0.496 (0.015)
	TAE-O	0.500	0.176	0.571	0.715 (0.040)	0.467 (0.039)
	TAE-SM80	0.625	0.088	0.769	0.757 (0.046)	0.615 (0.027)
	USNC-O	0.409	0.011	0.418	0.805 (0.037)	0.503 (0.043)
	USNC-SM50	0.545	0.070	0.490	0.810 (0.039)	0.516 (0.034)

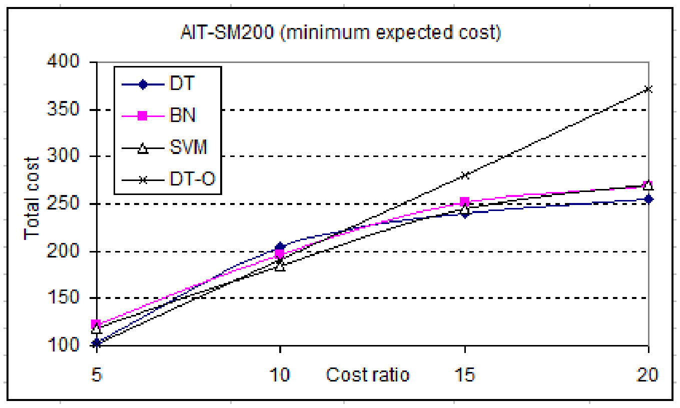

Figure 3. Total cost of AIT-SM200

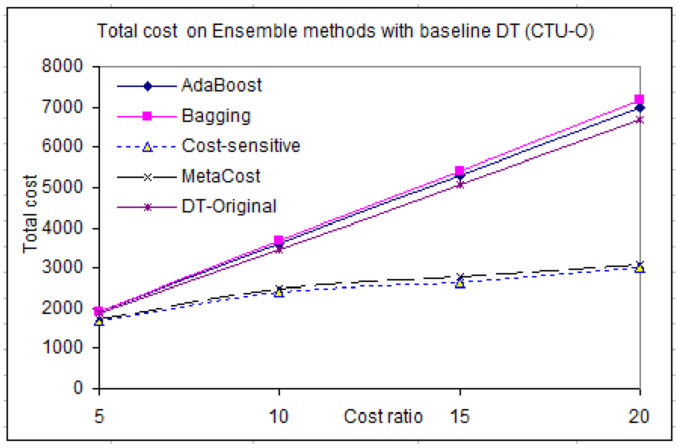

Figure 4. Total cost of ensembles (CTU-O)

total cost as a linear function when the cost ratio is increased. MetaCost is better than CSL in the case of small datasets (e.g. AIT-O), but when learning on larger datasets (e.g. CTU-O) then CSL outperforms.

VI. Discussions

In this study, the results have been improved when comparing with our previous works [6], which only worked on original datasets. Although some papers had researched in this area such as [3], [4], and [5], they evaluated their models in terms of accuracy, which has less meaning in the case of class imbalance as we analyzed in section 2.3. We have used AUC, F-Measure, and total cost to evaluate the models on three methods proposed in section 3 and recognized that the results are reasonable. When the datasets are large enough, CSL is better than MetaCost in the case of increasing cost ratio.

In the ethical issues, it should be aware that there is no segregation from using this system to treat with ”low performance” or ”high performance” students. As we said in the introduction, the purpose of this study is to improve the prediction results to help the them get better in studying.

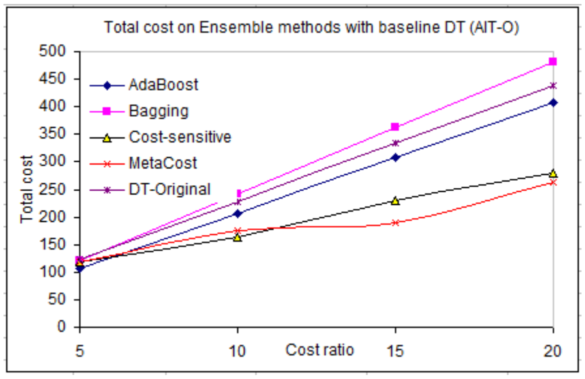

Figure 5. Total cost of ensembles (AIT-O)

VII. Conclusions

In this study, we have applied machine learning techniques to improve the prediction results of academic performances for two the real case studies. Three methods have been used to deal with the class imbalance problem and all of them show satisfactory results. We first re-balanced the datasets and then used both cost-insensitive and sensitive learning with SVM for the small datasets and with Decision Tree for the larger datasets. Our models are initially deployed on the local web. In future works, we will cross the results from using this system with the expected results in educational field.

In addition, we can integrate more features such as combining with E-learning system or predicting results in the E-learning. Moreover, most universities today are using the Credit System, in which each semester the students get confused with dozen of courses. So, how to help them choose appropriate courses based on their previous study results becomes the real problem. We can solve this by using association rules or recommender system approach [19] [20].

Acknowledgment

This research was funded by the ”Teaching and Research Innovation Grant” Project of Can Tho University-Vietnam, and partly supported by the Information Systems and Machine Learning Lab, University of Hildesheim, Germany.

References

[1] C. Romero, S. Ventura, P. G. Espejo, and C. Hervs, “Data mining algorithms to classify students,” *1st International Conference on Educational Data Mining*, June 2008.

[2] N. Delavari, M. R. Beikzadeh, and M. R. A. Shirazi, “A new model for using data mining in higher educational system,” *5th International Conf. on Information Technology Based Higher Education and Training*, 2004.

[3] R. Bekele and W. Menzel, “A bayesian approach to predict performance of a student (bapps): A case with ethiopian students,” *International Conference on AI and Applications*, 2005.

[4] S. B. Kotsiantis and P. E. Pintelas, “Predicting students marks in hellenic open university,” *IEEE Conference on Advanced Learning Technologies*, vol. 30, pp. 664–668, 2005.

[5] B. Minaei-Bidgoli, D. A. Kashy, G. Kortemeyer, and W. F. Punch, “Predicting student performance: an application of data mining methods with an educational web-based system,” *33rd Annual Conference on Frontiers in Education (FIE 2003)*, vol. 1, pp. 13–18, 2003.

[6] N. Thai-Nghe, P. Janecek, and P. Haddawy, “A comparative analysis of techniques for predicting academic performance,” *37th IEEE Frontiers in Education Conference*, pp. T2G7–T2G12, October 2007.

[7] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, “Handling imbalanced datasets: A review,” *International Transactions on Computer Science and Engineering*, vol. 30, pp. 25–36, 2006.

[8] N. V. Chawla, K. Bowyer, L. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” *Journal of AI Research*, vol. 16, pp. 321–357, 2002.

[9] C. Elkan, “The foundations of cost-senstive learning,” *17th International Joint Conference on Artificial Intelligence*, pp. 973–978, 2001.

[10] P. Domingos, “Metacost: A general method for making classifiers cost-sensitive,” *5th ACM SIGKDD International conference on Knowledge Discovery and Data mining KDD1999*, pp. 155–164, 1999.

[11] A. P. Bradley, “The use of the area under the roc curve in the evaluation of machine learning algorithms,” *Pattern Recognition*, vol. 7, no. 30, pp. 1145–1159, 1997.

[12] G. M. Weiss and F. Provost, “Learning when training data are costly: the effect of class distribution on tree induction,” *Journal of AI Research*, pp. 315–354, 2003.

[13] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd ed. Morgan Kaufmann, San Francisco, 2005.

[14] N. Friedman and M. Goldszmidt, “Building classifiers using bayesian networks,” in *Proceedings of the thirteenth national conference on artificial intelligence*, 1996, pp. 1277–1284.

[15] J. Cheng and R. Greiner, “Learning bayesian belief network classifiers: Algorithms and system,” in *AI ’01: Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence*. London, UK: Springer-Verlag, 2001, pp. 141–151.

[16] G. F. Cooper and T. Dietterich, “A bayesian method for the induction of probabilistic networks from data,” in *Machine Learning*, 1992, pp. 309–347.

[17] C. W. Hsu, C. C. Chang, and C. J. Lin, *A practical guide to support vector classification*, Department of Computer Science and Information Engineering, National Taiwan University, 2003.

[18] Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm,” *International Conference on Machine Learning*, pp. 148–156, 1996.

[19] A. Felfernig, G. Friedrich, and L. Schmidt-Thieme, “Intelligent systems special issue on recommender systems,” *IEEE Computer Society*, 2007.

[20] M. P. OMahony and B. Smyth, “A recommender system for on-line course enrolment: An initial study,” *ACM Conference on Recommender System*, pp. 973–978, October 2007.