Setting Up Automated Programming Assessment System for Higher Education Database Course

IGOR MEKTEROVIĆ, LJILJANA BRKIĆ
Department of Applied Computing, Faculty of Electrical Engineering and Computing,
University of Zagreb
Unska 3, Zagreb
CROATIA
igor.mekterovic@fer.hr http://www.fer.unizg.hr/en/igor.mekterovic
ljiljana.brkic@fer.hr http://www.fer.unizg.hr/en/ljiljana.brkic

Abstract: - The need for decreasing the workload among educators, timely feedback and accuracy and consistency on the grading results are the common reasons that motivate the development of Automated Programming Assessment Systems (APAS). Using our newly deployed APAS on Databases course we have evaluated students' knowledge using a "little and often" pattern and employed gathered data to predict students' performance on the course. The course's pass percentage and students' grades were predicted using several data mining techniques. Besides internal data gathered during course's execution, additional external variables (like grade point average) were considered for the data mining models. Our analysis shows that the accuracy of prediction is not highly affected if external variables are unknown. The pass percentage accuracy predictions is sufficiently high, especially after the half of semester (~80%), which allows for proactive approach towards the students we believe will fail the course. Other than that, we came up with few valuable insights into the structure and content of assignments that shall be applied in the next course cycle, in accordance with our intention to use course's data to improve the teaching and optimize the staff's man-hours.

Key-Words: Educational Data Mining, Academic Performance, Prediction, Automated Assessment

1 Introduction

Students' performance and course grades are of great importance to higher education institutions (HEI). Besides the need to educate a competitive workforce, HEIs are highly motivated to make their students achieve as good results as possible because the students' academic performance is the common criterion while evaluating the quality of a HEI [1]. One of the most popular techniques to analyze students' performance is data mining. Educational Data Mining is a discipline dealing with development of methods for discovering patterns within the data in educational databases, and using those methods to enlarge knowledge about educational phenomena and to comprehend students better, as well as the context which they learn in. Educational Data Mining has been used for improving the services HEI provide both for increasing students' grades and retention. We are interested in predicting students' performance on database course we are involved in to apply effective teaching approach and consequently change and improve student success. These efforts, although seemingly "small-scale", can have a large impact on student success. The research has shown that data describing students demographic characteristics (gender, age, disabilities, …) and data on academic achievements are essential in such prediction [2].

Since we have experience in teaching courses from computer science field, with a few hundreds of students enrolled we are very aware of difficulties in conducting and marking assessments for large classes. In many higher education institutions a disproportion between the number of students enrolled at a specific course and the number of staff involved is a common occurrence. This problem is especially present in courses where students need to acquire practical experience since this kind of knowledge imposes a multiple repetition of similar tasks in small groups or individually. Programming courses being taught to the students on computing studies are certainly such. When the marking for the large class sizes is done manually, educators' workload can grow to unmanageable extent. In manually marking environment, as a result of the time spent marking students' work, feedback is usually delayed. Also, when large numbers of educators are involved there can be a significant divergence in the standards applied when marking. The decreasing of the workload among educators can be achieved by increasing the number of educators or by minimizing the amount of manually marked assessments submitted by students. The overcome stated obstacles educators are turning to automated marking of students' assessments. Besides instant feedback to students, consistency and objectivity on the marking results are the common reasoning behind the need for Automated Programming Assessment Systems (APAS).

Automated assessment allows educators to evaluate students' knowledge using a "little and often" pattern, where the students produce many small-scale tasks during the course. It is especially convenient for the courses in computer programming where students need to apply programming constructs to solving a computational task. Like many other applied skills, practice and repetition are critical to becoming a skilled programmer. In the context of APAS students can perform as much practical work as possible in the time available. It is also essential that assignments are thoroughly designed and marked because the way and quality of the marking as well as overall grading in the course have a significant impact on students' motivation and learning approach.

Automated assessment enables gathering a substantial amount of data about education process and educators. We are using that data to predict and improve students' performance.

This paper describes the implementation of a few data mining models to predict pass percentage and students' grades on the course Database taught in the 4th semester of our bachelor study Computing. The models use data from institution's information system describing students demographic characteristics (gender), previous educational results (GPA, college admission points, …) and data gathered from our custom automated assessment system for marking two types of tests: (i) multiple choice test and (ii) SQL queries assignments.

2 Related work

Educational Data Mining researchers explore a variety of areas including educational software, computer- testing, computer supported collaborative learning and many other areas. Across these domains, one key area of application has been the investigating the factors that are associated with student failure or non-retention in courses or in university studies altogether [3] [4]. Different types of classification methods have been used to predict students' grades or scores. Usage of genetic algorithms to predict student final grade is show in [5]. Different data mining methods were used in [6] to predict a student's academic success.

Predicting students' grades: A, B, C, D, E and F using neural networks is presented in [7]; pass and fail using regression techniques in [8] or using neural network models on Moodle logs [9].

Also, there is a number of papers describing different aspects of using automated assessment in computer science courses [10] [11]. Automated assessment has been applied in programming courses [12] [13] [14] [15], data structures courses [16] [17], database courses [18], and system administration courses [19]. It has been used for marking various kinds of assessments, including programs with graphical user interfaces [20] [21], different types of diagrams [22] [23], SQL queries and other databases' concepts [24] [25]. In this paper we focus on the course Database taught in the 4th semester of our bachelor computing study.

2 Problem Formulation

Initially, we were motivated to develop an online automated assessment system due to lack of manpower, having considerable number of students enrolled in the database course (typically between 350 and 450) and only two teaching assistants. It was and still is our firm belief that written (paper) exams are the ultimate knowledge estimators in this domain, where a large body of code and written unstructured answers are inspected and evaluated by a human. However, online exams, could prove to be a worthy addition to the fundamental written exams. Our final goal is to introduce online exams as much as possible and thus reduce the volume of handwritten (paper) exams and hours spent assessing tests, without jeopardizing the overall marking assessment quality. Therein lies out first goal - to successfully structure our online grading system. In our first attempt, in the spring of 2017, we've introduced the following assignment structure, as shown in Table1.

Table 1 .Database course assignments structure

Ord.	Name	Type	Points
1	1st homework (SQL)	Online/u	1
2	2nd homework (SQL)	Online/u	1
3	1st multiple choice	Online/s	5.8
4	1st SQL (code) questions	Online/s	4.2
5	**Midterm exam**	**Paper/s**	**30**
6	3rd homework (SQL)	Online/u	1
7	4th homework (SQL)	Online/u	1
8	2nd multiple choice	Online/s	8
9	2nd SQL (code) questions	Online/s	3
10	Lecturer's points	Arbitrary	5
11	**Final exam**	**Paper/s**	**40**

In total, there are 100 possible points. Students with more than 50 points pass the course, with grades equally divided in the 50-100 range. Combined, online exams make 25 points (i.e. 25%). There are essentially two types of online exams: supervised and unsupervised (denoted with “u” and “s” in the Table 1). Unsupervised tests are written for an extended period of time (e.g. one week), typically at home. Supervised tests are taken for short period of time (e.g. one hour) in a laboratory under the teaching assistants’ supervision. Our online testing system supports two types of questions: multiple choice and SQL (code) questions. In the latter, a student is expected to write a SQL query for the given problem, which is then evaluated by running the query and comparing the result set with the (hidden) correct answer’s result set. Therefore, we have a system that allows us to question both theory (via multiple choice questions) and code. Still, certain topics cannot be properly addressed (e.g. draw a b-tree for the given data) in comparison to written exams. Although they make only 25%, there is a considerable number of online tests – eight tests are divided in two cycles, with written midterm exam positioned at the center of the semester. These fine-grained tests could allow us to more closely (and sooner than later) monitor students’ progress and act accordingly. Our second goal is then to develop a system to closely monitor students and be proactive, ultimately increasing the pass percentage. Students than do not pass the course in the continuous fashion (via these 11 test) take up to three more written and oral exams, increasing the man-hours on both sides. To achieve these goals, in the following sections we process the data gathered in the spring of 2017 to reflect on our testing structure and assess different data mining models predicting whether student will pass the course, and with which grade.

2.1 Data and data sources

We divide data in two categories: external and internal. External data is data external to the database course, that is, data accumulated before the course was enrolled. Internal data is data acquired during the course’s execution. Internal data is gathered through our APAS, and is at our disposal, while external data might generally be unavailable, depending on the higher education institution’s data information policy. Table 2 lists all the variables and their sources.

Table 2. List of all variables and their sources

Variable name	Description	Source
Gender	Male/Female	Ext
avgHS	Average high school grade (2-5)	Ext
HS	High school points (0-400)	Ext
SAT	Standardized college admission test (0-600)	Ext
sumHS	Total college admission points (0-1000)	Ext
rank	Student’s ordinal on admission (based on sumHS)	Ext
GPA	Grade Point Average for courses taken before enrolling Databases course	Ext
firstEnr	Whether student is enrolling the Database course first time (1st) or repeating (RPT) due to unsuccessful attempt	Ext
H1-HW4	Unsupervised homework one to four, see Table 1	Int
ABC1-ABC2	Supervised multiple choice tests one and two, see Table 1	Int
SQL1-SQL2	Supervised code (SQL) tests one and two, see Table 1	Int
me	Handwritten midterm exam	Int
~~fe~~	Handwritten final exam, not considered for prediction, since it is “too late”.	Int

2.2 Methodology

Dataset of 361 observations was divided into the train (75%) and test set (25%). Training set was used to evaluate four data mining models with their variations using 10-fold cross validation to select the appropriate model. Test set was used to evaluate model on unseen data. We have used R – a free software environment for statistical computing and graphics [26] (with several additional libraries) and RStudio, free and open-source integrated development environment for R. We have considered the following models:

(1) Decision tree and pruned decision tree: rpart package [27] for classification and regression trees was used. Default tree was subsequently pruned to avoid overfitting (the tree with least cross-validated error was selected)

(2) Random forest model with 2000 trees, also using rpart package
(3) Support vector machines: e1071 package was used. We have evaluated linear kernel, radial kernel, and tuned radial SVM to find the best cost (varied from 10^{-2} to 10^{3}) and gamma (varied from 10^{-3} to 10^{3})
(4) Logistic regression using glm for binomial and nnet package for mulinomial classification

3 Results and discussion

We have assessed models for three different variable groups with regards to different points in time:

- External variables – known at the time of the enrollment, before the first lecture
- Internal variables – accumulated during the course's execution, with and without the written midterm exam
- All variables – known at the end of course, before the final exam

Courses' pass percentage (in a continuous fashion) in the year 2017 was 60.52% and the most common grade was Fail (1) with the remaining 39.48% which constitutes a naive baseline model for the comparison. Pass percentage is of our primary concern, and pass percentage was used to assess models. We have also predicted grades, as an additional feature, but with minor priority.

The following tables 3-6 show the average accuracy and standard deviation σ of our data mining models obtained through 10-fold cross validation with regards to different variable groups. Best performing models are denoted with asterisk, and chosen models are denoted with double-asterisk. In the spirit of the "one-standard-error" rule [28], we are inclined towards more simple, potentially glass-box models.

Table 3. External variables, 10-fold cross validation average accuracy

Model	Pass		Grade	
	Accur	σ	Accur	σ
Dec. Tree**	0.7259	0.0634	0.4889	0.0777
Pruned Dec Tree*	0.7333	0.1000	0.5815	0.1105
Random Forest (N=2000)	0.6778	0.1019	0.4704	0.0631
SVM, linear kernel	0.7148	0.0838	0.4815	0.1145
SVM radial kernel, default	0.6889	0.0634	0.4741	0.0834
SVM radial kernel tuned*	0.7185	0.0804	0.4481	0.0947
Log regression	0.6889	0.0584	0.4185	0.0631

All models perform in the 67%-74% average accuracy range, SVM and (pruned) decision tree as best among them. Decision tree was selected for the model, because it is a simple and interpretable model. Images 1 and 2 shows the decision trees for default and pruned decision tree. Note that pruned tree considers only one variable – average grade (attained in the first year of study), while the default tree considers additional variables from the admission (e.g. HS). Default tree proved slightly better on the validation data (69% > 67%).

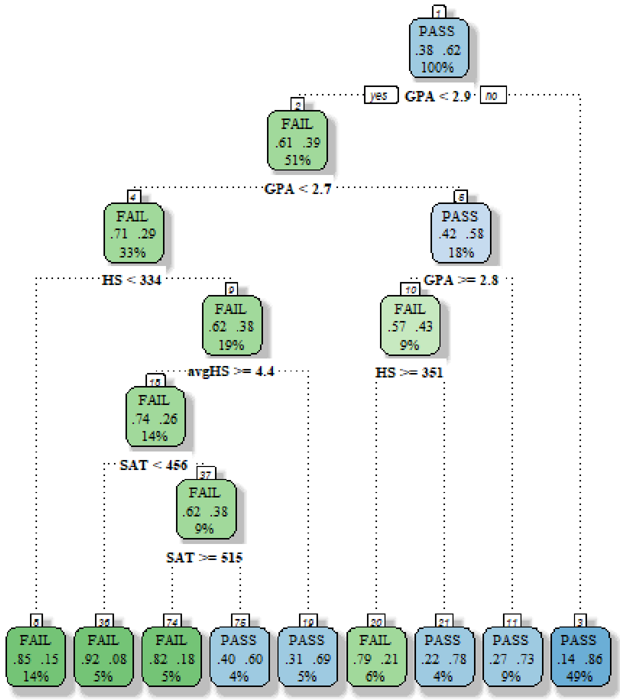

Figure 1. Decision tree predicting pass for external variables

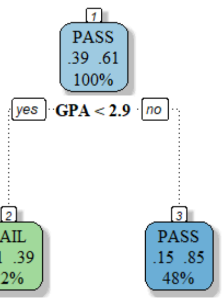

Figure 2. Pruned decision tree from Fig1

Table 4 shows the results for the internal variables:

Table 4. Internal variables, without midterm exam, 10-fold cross validation average accuracy

Model	Pass		Grade	
	Accur	σ	Accur	σ

Decision Tree	0.6926	0.0580	0.3778	0.0383
Pruned Dec. Tree	0.7593	0.0659	0.4815	0.0653
Random forest (N=2000) ***	0.8000	0.0584	0.4333	0.0891
SVM, linear kernel	0.7519	0.0606	0.4852	0.1164
SVM radial kernel, default	0.7185	0.0804	0.4296	0.0841
SVM radial kernel tuned	0.7704	0.0904	0.4852	0.0750
Log regression	0.7741	0.0931	0.5111	0.0815

Random forest shows best results when midterm exam is not considered (and logistic regression as the grade estimator). We have performed this analysis to gauge the correlation of online tests to the overall pass percentage and grade, and gain insight into our test structure. It shows that we can estimate the pass percentage with 70+% accuracy with online tests taking up 25% of the overall points. When inspecting the log regression model for variable significance (z-statistic) the following variable are singled out: SQL1, ABC2, ABC1 and HW4. Random forest considers similar variables - Figure 3 provides a plot of mean decrease in accuracy and GINI (the average gain of purity by splits of a variable):

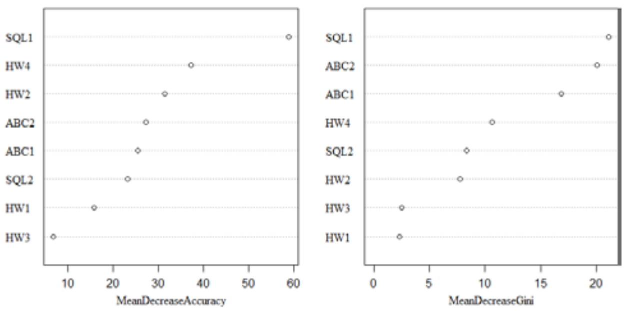

Figure 3 Mean decrease in accuracy and Gini for the random forest model, midterm excluded

Also, Fig 4 shows the decision tree for internal variables:

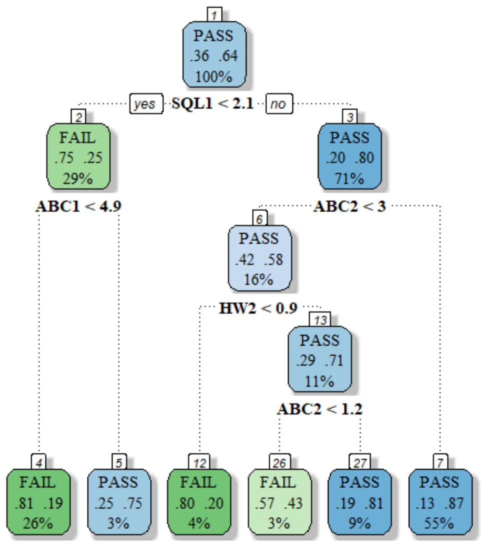

Figure 4. Decision tree for the internal variables

Finally, if we simply calculate correlation coefficients on the entire data set:

Table 5. Correlation of internal variables with the pass variable, 10-fold cross validation average accuracy

pass vs:	HW1	HW2	HW3	HW4	ABC1	SQL1	ABC2	SQL2
r	0.15	0.23	0.20	0.31	0.29	0.48	0.39	0.30
p-value	4e-3	7e-6	1e-4	1e-09	1e-08	2e-16	3e-14	1e-8

The first SQL code test (SQL1) is consistently the strongest pass predictor, and ABC1 and ABC2 are also considered in all models. SQL2 appears less significant than we thought, probably due to the small amount of points awarded (Table 1), and we shall correct that in the following year. Never the less, SQL test seem more significant than their points-value, which leads us to believe that SQL questions are more important than multi-choice question. Since both SQL1 and ABC1 occur early in the semester (before midterm) this gives us an opportunity to act sooner in the semester. Surprisingly, homework 4 and 2 play a part in the models, and homework 1 and 3 are insignificant. Taking into the account that each homework weight only 1% of the total points, we have expected them all to be insignificant. The intent of the homework was to keep students engaged, and to follow the course. The testing system actually tells students whether their homework solution is correct (before they submit it) and they can try to correct it for as many times as they want. So, for the homework part, the student knows how much points he or she will get, before submitting. When inspecting more closely the homework distribution, we notice that homework 2 and 4 indeed differs from 1 and 3 (Fig. 5).

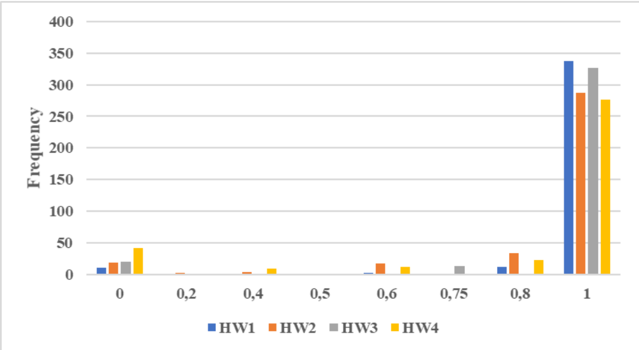

Figure 5. Homework 1-4 histogram

Homework 2 and 4 are more selective than 1 and 3, probably because they are harder (average scores being 0.96, 0.90, 0.93 and 0.84 respectively). We believe that this is due to the fact that more perseverant students and those intrinsically interested in the subject will pursue the solution regardless the fact it carries very small reward, while students on the other side of the spectrum will probably give up on hard questions thinking that they are not worth the effort. Therein lies a valuable insight: easy homework is pointless, and hard homework, when given enough time, can be a valuable indicator even though it carries a small points percentage!

On another topic, out intent to find students about to fail and act accordingly, there is no reason not to include the midterm exam variable, since it is an internal variable known to the system at the middle of semester. Of course, midterm exam will take the first place among the predictors and improve the pass prediction accuracy, as show in the Table 6 and Figure 6:

Table 6. Internal variables, with midterm exam, 10-fold cross validation average accuracy

Model	Pass		Grade	
	Accur	σ	Accur	σ
Decision Tree	0.8111	0.0564	0.5704	0.0927
Pruned Dec. Tree	0.8444	0.0547	0.5741	0.1066
Random forest (N=2000)**	0.8556	0.0750	0.6444	0.1006
SVM, linear kernel	0.8333	0.0531	0.637	0.0777
SVM radial kernel, default	0.8593	0.0672	0.6037	0.0957
SVM radial kernel tuned*	0.8778	0.0464	0.6222	0.1059
Log regression	0.8333	0.0895	0.6444	0.1093

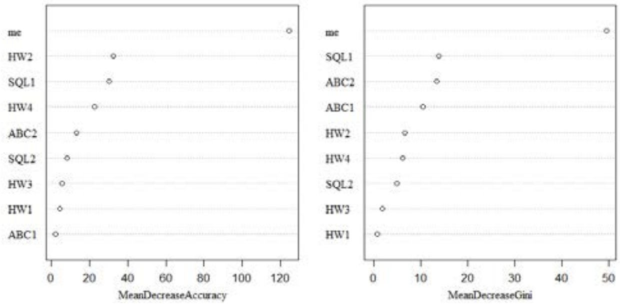

Figure 6. Mean decrease in accuracy and Gini for the random forest model, midterm included

At the beginning of the second cycle in the semester, we can predict whether student will pass with roughly 80% accuracy, and grade with 60% accuracy, using only course's internal variables (e.g. random forest model scored 78% and 55% on the unseen data).

Finally, we perform the model comparison for all variables, to assess the impact of external variables:

Table 7. Internal and external variables, with midterm exam, 10-fold cross validation average accuracy

Model	Pass		Grade	
	Acc	σ	Acc	σ
Decision Tree	0.8000	0.0584	0.6148	0.0859
Pruned Dec. Tree	0.8333	0.0659	0.6778	0.0580
Random forest (N=2000)*,**	0.8926	0.0616	0.7000	0.0537
SVM, linear kernel	0.8444	0.0649	0.6259	0.0845
SVM radial kernel, default	0.8593	0.0694	0.6037	0.0925
SVM radial kernel tuned	0.8852	0.0564	0.6741	0.0796
Log regression	0.8296	0.0358	0.6778	0.0580

Overall accuracy is slightly improved. Random forest scored 79% and 56% on unseen data. On a more positive note, prediction model will not work significantly worse if external variables are not known, which, in general, might easily be the case. External variables can be divided into GPA + others, where GPA is the most significant and most likely to be known since it belongs to the same (higher) education institution. Other external variables considered here are gathered during the high school and in the admission process, except for gender and firstEnr which proved insignificant in our case. Even if GPA is not known to the system, students could be asked by the system to volunteer that information, to their benefit.

4 Conclusion

In this paper we have examined the impact of our newly deployed online testing system on the course's pass percentage and students' grades. Several potential data mining models were evaluated to find the most fitting binomial classificator, i.e. to determine, the sooner the better, whether a student will pass the course or fail. Predictor variables were divided into two groups - variables external to the automated assessment system (course) and variables acquired throughout the course's execution. In the process, we have acquired several insights relating to the testing structure and impact, briefly:

- SQL (code) questions are more valuable (selective) than multiple choice questions. Our SQL2 exam was poorly valued.
- Unsupervised online homework should be hard, even if it carries a very small reward in points. Students must have enough time to solve it (e.g. a week)
- We can predict with reasonable accuracy, especially after the midterm exam (~80%) who will pass and who will fail. In other words, "little and often" patterns provides valuable data during the course's execution to act on it.
- Predictions do not deteriorate significantly if external variables are unknown (i.e. if we predict solely using internal variables)

In our future work we shall apply these conclusions to the next course instance, and reassess the situation.

References:

[1] B. Wächter, M. Kelo , Q. Lam, P. Effertz , C. Jost and S. Kottowski, "University Quality Indicators: A Critical Assessment," 2015.

[2] A. M. Shahiri, H. Wahidah and A. R. Nur'aini, "A Review on Predicting Student's Performance using Data Mining Techniques," in *The Third Information Systems International Conference*, Surabaya, Indonesia, 2015.

[3] G. W. Dekker, M. Pechenizkiy and J. M. Vleeshouwers, "Predicting students drop out : a case study," in *Proceedings of the 2nd International Conference on Educational Data Mining, EDM*, Cordoba, Spain, 2009.

[4] C. Romero, S. Ventura, C. Hervás and P. G. Espejo, "Data Mining Algorithms to Classify Students," in *Proceedings of the 1st International Conference on*, Proceedings of the 1st International Conference on, 2008.

[5] B. Minaei-Bidgoli and W. F. Punch, "Using Genetic Algorithms for Data Mining Using Genetic Algorithms for Data Mining," in *Genetic and Evolutionary*, 2003.

[6] J. F. Superby, J. P. Vandamme and N. Meskens, "Determination of Factors Influencing the Achievement of the First-year University Students using Data Mining Methods," in *Workshop on Educational Data Mining*, 2006.

[7] L. V. Fausett and W. Elwasif, "Predicting Performance from Test Scores using Backpropagation and Counterpropagation," in *IEEE World Congress on Computational Intelligence*, Orlando, FL, USA, 1994.

[8] S. B. Kotsiantis and P. E. Pintelas, "Predicting Students Marks in Hellenic Open University," in *Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05)*, Kaohsiung, Taiwan, 2005.

[9] M. D. Calvo-Flores, E. Gibaja, M. C. Pegalajar Jiménez and O. Pérez, "Predicting Students' Marks from from Moodle logs using neural network models," in *Current Developments in Technology-Assisted Education*, 2006.

[10] J. English and T. English, "Experiences of Using Automated Assessment in," *Journal of Information Technology Education: Innovations in Practice*, vol. 14, pp. 237-254, 2015.

[11] R. S. Pettit, J. Homer, K. Holcomb, S. Nevan and S. Mengel, "Are automated assessment tools helpful in programming courses?," in *122nd ASEE Anual Conference & Exposition*, Seattle, Washington, 2015.

[12] M. Hull, D. Powell and E. Klein, "Infandago: Automated Grading for Student Programming," in *Proceedings*, Darmstadt, Germany. June, 2011.

[13] V. Pieterse, "Automated Assessment of Programming Assignments," in *Proceedings of the 3rd Computer Science Education Research Conference on Computer Science Education Research*, Arnhem, Netherlands — April 04 - 05, 2013 , 2013.

[14] R. Singh, S. Gulwani and A. Solar-Le, "Automated feedback generation for introductory programming assignments," in *Proceedings of the 34th ACM SIGPLAN conference on*, 2013.

[15] R. Romli, S. Sulaiman and K. Z. Zamlic, "Improving Automated Programming Assessments User Experience Evaluation Using FaSt-generator," in *The Third*

Information Systems International Conference, 2015.

[16] W. Siever, "Automated assessment in data structures: a summary of experience," *Journal of Computing Sciences in Colleges*, vol. 30, no. 2, pp. 17-22, 2014.

[17] M.-J. Laakso, T. Salakoski, A. Korhonen, and L. Malmi, "Automatic Assessment of Exercises for Algorithms and Data Structures – a Case Study with TRAKLA2," in *Proceedings of 4th Finnish/Baltic Sea Conference on Computer Science Education*, 2004.

[18] J. C. Prior and R. Lister, "The Backwash Effect on SQL Skills Grading," in *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education*, Leeds, United Kngdm — June 28 - 30, 2004 , 2004.

[19] L. Baumstark and E. Rudolph, "Automated online grading for virtual machine-based systems administration courses," in *Proceeding of the 44th ACM technical symposium on Computer science education*, Denver, Colorado, USA, 2013.

[20] A. McAllister and M. Yu Feng , "Automated grading of student-designed GUI programs," in *World Transactions on Engineering and Technology Education*, 2005.

[21] J. English, "Automated assessment of GUI programs using JEWL," in *Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, Leeds, UK, 2004.

[22] V. Vachharajani, J. Pareek and S. Gulabani, "Effective Label Matching for Automatic Evaluation of Use -- Case Diagrams," in *2012 IEEE Fourth International Conference on Technology for Education (T4E),*, Hyderabad, India, 2012.

[23] P. G. Thomas, "Online Automatic Marking of Diagrams," *Systemic Practice and Action Research ·*, 2013.

[24] B. Chandra, M. Joseph, B. Radhakrishnan, S. Acharya and S. Sudarshan, "Partial marking for automated grading of SQL queries," *Proceedings of the VLDB Endowment*, vol. 9, no. 13, pp. 1541-1544, 2016.

[25] S. Dekeyser, M. De Raadt and T. Yu Lee, "Computer Assisted Assessment of SQL Query Skills," in *Eighteenth Australasian Database Conference (ADC2007)*, Ballarat, Australia, 2007.

[26] "The R Project for Statistical Computing," [Online]. Available: https://www.r-project.org/. [Accessed 2017].

[27] "rpart: Recursive Partitioning and Regression Trees," [Online]. Available: https://cran.r-project.org/web/packages/rpart/index.html. [Accessed 2017].

[28] D. Krstajic, . L. J. Buturovic, D. E. Leahy and S. Thomas, "Cross-validation pitfalls when selecting and assessing regression and classification models.," *Journal of Cheminformatics,* 2014.