Predicting students drop out: a case study

Gerben W. Dekker
g.w.dekker@student.tue.nl
Department of Electrical Engineering,
Eindhoven University of Technology

April 10, 2009

Abstract[1]

In the emerging field of educational data mining, a strong bias towards data-rich digital learning environments is the current state of affairs [2, table 2]. However, in many educational institutes a lot of regular course data will probably be more readily available. This data may also be used to support and advise students in various ways, for the better of the student as well as the institute.

In this study, the situation at the Electrical Engineering department of Eindhoven University of Technology is considered. Based on experience, the department claims to be able to distinguish the potentially successful students from amongst the first year influx before the end of the first semester. To do this in an early stage is important for the student as well as for the university, but the selection is only loosely based on assumed student similarities over the years. There is no thorough analysis. Data mining techniques may corroborate and improve the accuracy of this prediction. Furthermore, data mining techniques may point out indicators of academic success that are missed until now.

This study explores if and how data mining techniques can be applied in this practical situation. The techniques are applied on data that is readily available in the institution's database. In contrast to some other studies [3], no additional data is collected to make the results of the applied techniques easily applicable.

[1] This is the extended technical report, accompanying the paper under the same title [1]. The report describes the data mining process in detail, including preprocessing data-related issues, and is intended to be accessible for a general public. The paper concentrates more on the results obtained and is written for domain experts.

This report is the final result of an internship by the author at the Department of Computer Science, Eindhoven University of Technology, carried out as part of the Master's program Electrical Engineering.
The intership was supervised by Mykola Pechenizkiy (Assistant Professor at the Department of Computer Science) and Jan Vleeshouwers (Student Counsellor at the Department of Electrical Engineering).

Acknowledgments

I would like to thank the Electrical Engineering Department of Eindhoven University of Technology kindly for providing me access to the department's database, and the trust placed in me by doing this.
Much thanks to my supervisors: Jan for taking the time to explain the endless lists of abbreviations in the database, and Mykola for supporting me throughout doing the research, being patient when I was learning SQL and helping me out in writing scripts to export the data.
Finally, also a lot of thanks to Ruth, who really pulled me through in the tedious preprocessing stage and afterwards.

Contents

1 Description of the situation

In the electrical engineering department of Eindhoven University of Technology, the monitoring and support of first year students is a topic that is considered very important. The department's yearly student enrollment for the Bachelor's program is lower than desired, and the curriculum is not an easy one. The drop out rate of freshmen is about 40%.

Apart from the institutional aim to enforce an upper bound to the drop-out rate, there are two reasons for the department to want to identify successful and unsuccessful students in an early stage. First there is the legal obligation the department has in providing students with the necessary support to evaluate their study choice. In general, students who choose to pursue their study career at another institution, should do this at an early stage. For Electrical Engineering students there is a very concrete reason to evaluate before the end of the first semester: the Electrical Engineering program of the nearby polytechnic Fontys Hogeschool accepts university drop outs in their curriculum until the beginning of January, without any time losses involved.

A second reason is that there is always a subset of students which the department considers a "risk group", i.e. students who may be successful but who need extra attention or specific individual care in order to succeed. Detecting this risk group in an early stage is essential for keeping these students from dropping out. It enables the department to direct its resources to the students who need it most.

To support students in making this decision, every enrolled student receives a study advice in December. This advice tells the student whether or not he or she is encouraged to proceed his study career at the faculty. It is based upon the grades and other results of the student so far and upon information obtained from 1st-semester-teachers and student-mentors, examined and interpreted by the department's student counselor. The final semester examinations are not taken into account, because they are in January; postponing the advice until after the results are known would preclude students from switching to Fontys Hogeschool. The advices seem to be quite accurate in practice: students who are assessed as potentially successful are in general the same students that are successful after a year. Moreover, the students who are not encouraged to proceed their current study program, generally do not continue into the second year.

Despite the success, the assessment remains unsatisfactory because of its rather subjective character. The department feels that:

1. A more robust and objective founding of the process may lead to advices which are more consistently followed up by students;
2. A closer analysis is likely to lead to an improved selection process.

First of all, the department is interested in which of the currently available student data are the strongest predictors of success, and in the performance of this predictor. The lower the predictor's quality, the more the department is curious to know what information makes the current assessment work. If the predictor quality is high, the department's interests are directed towards:

* Using the predictor as a back-up of the current assessment process;
* Identifying success-factors specific to the Electrical Engineering program;
* Identifying what data might result in a further increase of the predictor quality, and as a consequence, collect these data;
* Modifying the assessment quality, resulting in a better prediction, e.g. a more differentiated view on the risk group;
* Modifying the assessment process time-line, resulting in an earlier prediction, ideally even before entering the study.

Furthermore, if strong predictors for academic success can be found, these will also be used to gain understanding of success and risk factors regarding the curriculum. Awareness of these factors by teachers, education personnel and management will help to select appropriate measures to support the risk group, eventually resulting in a decrease of the drop-out rate.

2 Review of literature

The topic of explanation and prediction of academic performance is widely researched. It is impossible to assess the literature in depth in this short study. Nevertheless, a quick review will be given in this section. Firstly, the literature concerning identification of success in a traditional way (i.e. using multinomial regression or something similar) is reviewed. Secondly, the few studies considering data mining in this field are covered.

In the older studies, the model of Tinto [4] is the predominant theoretical framework for considering factors in academic success. Tinto considers the process of student attrition as a sociopsychological interplay between the characteristics of the student entering university and the experience at the institute. This interaction between the student's past and the academic environment leads to a degree of integration of the student into this new environment. According to this model, the higher the degree of integration, the higher the commitment to this institute and to the goal of study completion. Later studies tried to operationalize this model by trying to measure "integration". The study of Terenzini [5] came up with the following five factors, consisting of 34 items: Peer Group Interactions, Interactions with Faculty, Faculty Concern for Student Development and Teaching, Academic and Intellectual Development, and Institutional and Goal Commitments [5, p. 113]. These factors proved to have a predictive capacity across different institutions, and showed therefore to be a potential tool in identifying students who might drop out.

Other studies tried to identify the significant factors in a more detailed way. Many studies included a wide range of variables, including personality factors, intelligence and aptitude tests, academic achievement, previous college achievements, demographic data etc. [6, 7, 8, 9] Depending on the study, some factors came out stronger than others. Taking [8] as recent and European example, it showed the following significant factors in dropping out:

* sex (only in technical schools);
* age at enrollment;
* score on pre-university examination;
* type of pre-university education (vocational or secondary general education);
* whether or not enrolled in preferred course;
* type of financial support;
* father's level of education;
* resident of the university town or not.

What is clear from all studies is that academic success is dependent on many factors, where grades and achievements, personality and expectations, as well as sociological background all play a role.

The use of data-mining techniques in this field is relatively new. If data-mining is used in an educational context, it is mostly on virtual or web-based courses or distance learning [2]. It is not sure yet whether data mining can outperform classical methods in predicting academic success [10]. Furthermore, it is not clear which data mining algorithms are preferable in this context: [10] uses neural networks and decision trees, [11] uses these too and prefers decision trees, but also suggests clustering algorithms as data exploration tool, and [12] uses decision trees and Bayesian Networks. Neural networks seem to be suitable for big datasets especially [10, p.18], but has as disadvantage that the resulting model is not always easy to interpret. Decision trees, however, are easy to understand and interpret [13]. In [12], Bayesian networks are consistently outperformed by the decision tree algorithms. Another tool available is association analysis. However, it might well be that association rules are overrated, see [14]. Combining these results, it is clear that decision trees seem to be the a priori preferred method given our rather small data set, but literature is too scarce to conclude upon this.

There are some studies in which data mining is used to achieve a goal that is comparable with ours. Their findings are summarized here for reference. While the differences with the current

work are too substantial to do a detailed comparison, it gives a rough idea about the results to be expected.

* Herzog [10] tries to predict freshmen retention using data until the end of fall of the first year. He uses a dataset of 8018 students with 40 attributes, including pre-university grades, student demographics, campus experience and financial information. He finds a prediction accuracy of around 75%.
* Luan [11] uses a dataset with comparable attributes and 32,000 students. Using decision trees, an accuracy of 80-85% is found for binary classification.
* Thai Nghe et al. [12] uses admissions information to predict GPA at the end of the first year. An accuracy of 93% is reported for binary classification, using a dataset of 936 students with 14 attributes. A classification into three classes reaches an accuracy of 74%.

3 Methodology

The case study will consist of different stages, roughly following the cross-industry standard procedure CRISP-DM [15, p.6].

Firstly, the business understanding phase has to be carried out. In this phase, the project objectives and requirements are stated and refined and the resulting data mining problem is formulated. The results of these phase are summarized in the previous sections. An important aspect of the situation is that no additional data will be collected. Although the collection of additional data (especially suited for data mining) results in a richer data set and is therefore likely to give better results, a model acting on a data set that is already automatically kept up-to-date is potentially a much more useful tool.

In the data understanding phase, the data is collected and explored. The institution's database is explored and assessed on usefulness for the project. It has to be verified whether the data is rich enough, has enough relevant entries and is in a format that can be made suitable for data mining. Meanwhile, the insight in the data should be increased. The student counselor will be consulted during this process. This phase will be described in section 4.

In the data preparation phase, the initial raw data is transformed to a format suitable for mining, as described in section 5. This involves a careful data selection, data transformation and aggregation, and conversion of data formats into a format suitable for the used mining application. Furthermore, in this phase a suitable classification variable should be defined and its value should be derived from the data.

After these preparatory steps, the data is ready for mining and the modeling phase starts. In section 6 different modeling techniques will be selected and applied, and compared with each other. The obtained results are enhanced using classification weights, a process described in section 6.5. The modeling will take place using `Weka` [16], an open source collection of data mining algorithms.

The resulting model is evaluated in section 8 and its quality and errors are assessed. Finally, the results are discussed in section 9, and a number of recommendations will also be included there.

4 Data understanding

The information available for mining as a subset of the institution's database called OWIS. This OWIS database contains data about all the students enrolled or previously enrolled in one of the university courses. It also includes course details, course schedules, information about teachers, internships, exams and examinations, and graduation details. The student counselor of Electrical

Engineering has access to a small subset of this database, containing all data regarding courses, teachers and students who are related to one of the curricula of the department. This subset is retrieved as a set of 45 uncoupled tables, with table size ranging from only 1 record to about 110,000[2]. Many of the attributes were abbreviations which needed to be clarified by the student counselor. The subset was provided as a `Microsoft Access 2000` database.

During data exploration many data anomalies were discovered, like student ID's without students, people enrolled in many programs at the same time, missing values and so on. The author's knowledge about the department's real situation and the expertise of the student counselor proved to be invaluable in making sense of the data.

The richness of the data was less than hoped for. Concerning university grades all needed information was available: the exam date and the result of every attempt. But concerning pre-university education the results were mixed. If a student took the standard Dutch pre-university secondary education, most of the time all final grades were available. In the Dutch education system, there is however a second standard way into university, following one year of Electrical Engineering polytechnical education first. Of these students (approximately 10% of the bachelor influx), the secondary education results were not always available. For students entering the Bachelor's program via yet other routes, data was barely available.

More severe was the lack of any data other than course information. As stated in [3], Parmentier [6] showed that three sets of factors are of importance: the personal history of the student, the involvement in and behavior in relation to his studies (like participation, asking of questions etc.), and the way in which the student perceives his study environment (courses, professors, etc.). Based on the OWIS database, nothing can be said about social or psychological factors of students. We do not even know whether the student lives with its parents our not, or how far the student has to travel (both attributes that are intuitively potential indicators of success). This tempers our expectations of the learning possibilities of this dataset.

The correlation in the dataset was also investigated using a dendrogram, which shows correlation between students. It can be an indicator for existing clusters in the student population. The result is shown in figure 1. An immediate observation can be made concerning the cluster in the top right of the figure. The attributes listed there are all related to the partial examinations, which are introduced in 2007. All students that started before 2007 will not have entries in these fields, and all students starting after 2007 will at least have some results listed here. These students form a clearly defined cluster. A further analysis of this dendrogram is not been made due to the fact that it became available late in the process. It might be a very good starting point for subsequent projects though.

5 Data preparation

After this preliminary data exploration and understanding, the dataset to be mined had to be determined. The first step was to select exactly the right group of students. The important criterion is here to make sure that only students who really belong to the current bachelor curriculum are included. After some trial and error the following criteria where determined and coded into an SQL query:

* The student must be enrolled in a curriculum with a code associated with a Bachelor's degree of the electrical engineering department;
* The student must be in its first year phase, called the 'propedeuse' phase in the Dutch system;
* This enrollment must also be the first enrollment of this student at this university[3].

[2]It is clear that these tables are related in the original database. However, apparently these relations are not preserved when the subset is taken.

[3]This is necessary to prevent students who started in an earlier program and are transferred to the new curriculum for administrative reasons.

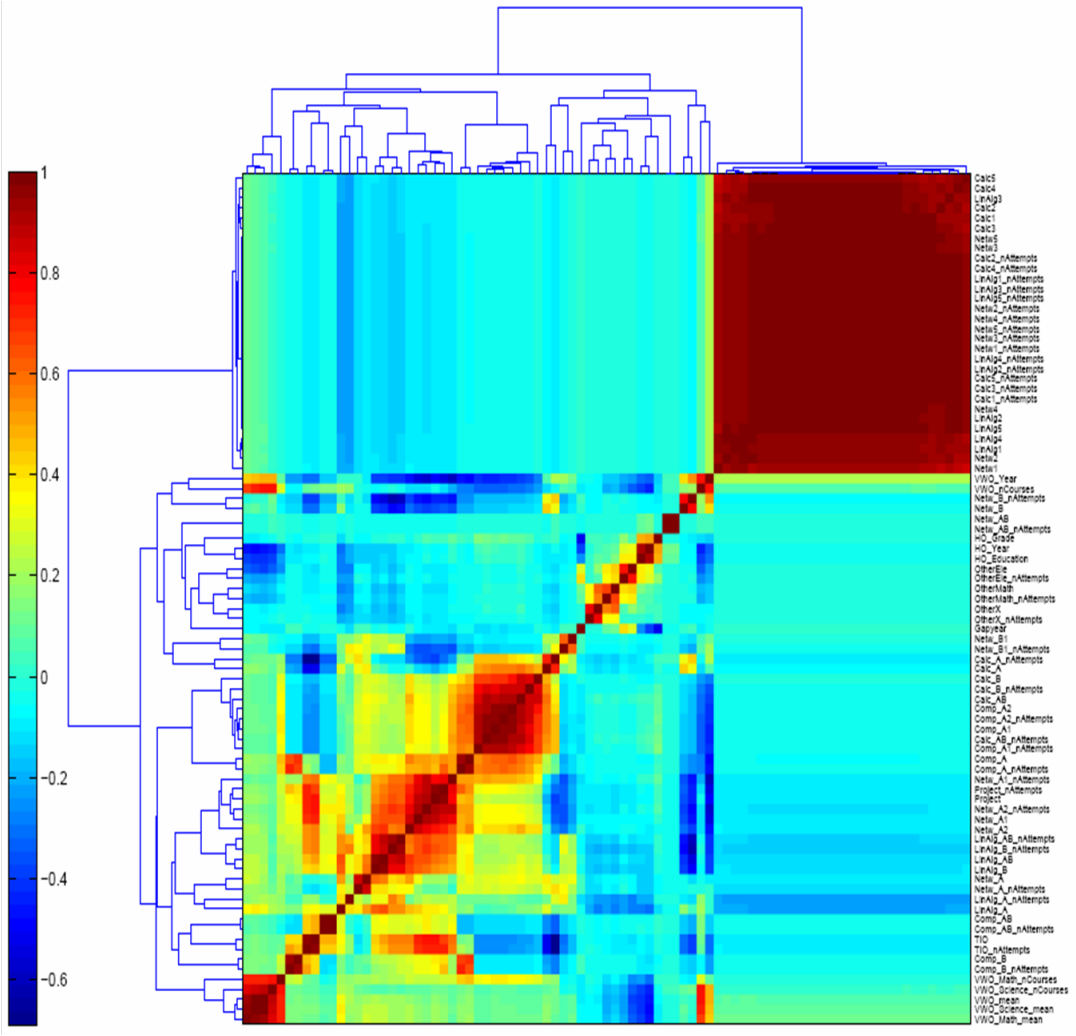

Figure 1: Dendrogram clustering results

This resulted in a set of 766 students. After further analysis also the exchange students and the wrongly classified master students were removed from the data, resulting in a smaller set of 648 students.

The data that is mined in these study can be divided into two sets: a dataset containing the attributes about the student's past education, and a dataset containing the university grades and related data. The composition of these datasets are described below.

5.1 The past education dataset

As stated shortly in section 4, there are different types of education a student could have followed before starting at university. The most common is to have had pre-university secondary education, called 'VWO' in Dutch. Of these students we have the most information. There might be data available of students with a different degree. These data should also be represented.

For VWO students, all examination grades are available. To treat all the grades as different attributes creates a very diverse set for the data mining algorithms. Instead, we choose to aggregate the information in less attributes. For the grades, we choose to lump them into six attributes: the total number of courses taken, the unweighted mean of these courses, the number of science courses

taken and their mean, and finally the number of mathematics courses taken and their mean. It remains to be seen how useful it is to take science and mathematics apart because they might be strongly correlated, but it is sometimes stated that the mathematics grade alone will be a strong predictor for success in this curriculum. Therefore we keep it apart. Furthermore, the attributes are set as nominal attributes in stead of just using numerical attributes. This might improve the mining classifiers. The grades are all classified in the same way. In the Dutch grading system a grade between 1 and 10 is given, with 10 as best result. The conversion is shown in table 1.

Grades	0	<6	6-6.5	6.5-7.5	7.5-8.5	>8.5
Category	n/a	poor	average	above average	good	excellent

Table 1: grade to category conversion

The other VWO attributes used are *VWO_Year* and *VWO_Profile*. These contain the year of graduation as well as the chosen predefined combination of courses, which is a feature of the Dutch education system. For students with another degree, the education type is listed (*'HO_Education'*) as well as the graduation year and grade.

Lastly, there is one extra attribute added, consisting of the difference in years between the graduation data of the last pre-university education of the student and the start of its enrollment at the department. By far the most students score '0' here, but some deviations do occur. A negative number means that the student has to finish some of his previous education during his university studies.

The attributes in this data set are exemplified and summarized in table 2.

5.2 The university grades dataset

Considering the university grades of the students, we limit ourselves of the results before December 31th of the enrollment year of the student. This is done because the department wishes to identify the successful students before Christmas (see section 1), so the learning scheme cannot take more data into account. In the first attempt of examining this data we ran into problems: every exam result is classified as belonging to a certain course, which is represented by a course code. In the first dataset we found more than 80 different course codes, where one would expect only a few due to the fact that all these students are supposed to follow more or less the same curriculum. Investigation showed that many courses have much more than one course code assigned to it. This is due to the fact that these courses have many subtypes. In fact, if it is possible to do a partial examination for a course, every part has a different course code. Furthermore, courses tend to be rearranged frequently. In this project, data from 2000 until 2009 is covered, resulting in a lot of rearrangements underway. The major tendency is to combine different courses into one course.

At the same time however, partial exams are carried out more frequently. The rationale behind this is that it is profitable for students to get feedback about their level of understanding of the course contents as soon as possible. If a student makes these partial examinations sufficiently, the final examination can be skipped. In that case, the final grade is the mean of the partial examinations. But if the partial examinations are not graded high enough, the student has to take the final examination. For the Calculus, the Linear Algebra and Electrical Networks courses the number of partial examinations is increased from two to five in recent years, giving even more grade variants.

These constant changes in examination practices makes it nontrivial how to combine the different examination results into a single course metric. Therefore, they are kept separate. There is an attribute noticing how many attempts are taken for a certain examination, and what the highest attained grade of these attempts was. This results in a dataset with $2 * 37 = 74$ attributes. Contrary

Attributes	Type	Typical value	Remarks
IDNR	numeric	0553453	Is removed after importing, used to check data integrity
VWO_Year	nominal	2001-2003	Five categories including 'n/a', corresponding to major changes in the Dutch education system
VWO_Profile	nominal	VWO(N+T)	The curriculum taken at pre-university education. Six categories including 'n/a'
VWO_nCourses	nominal	6	The number of courses taken. Integer values.
VWO_mean	nominal	'good'	{ n/a, poor, average, 'above average', good, excellent }
VWO_Science-_nCourses	nominal	>3	{ n/a, <3, 3, >3 }
VWO_Science-_mean	nominal	'good'	As VWO_mean
VWO_Math-_nCourses	nominal	1	{n/a, 0,1,2}
VWO_Math-_mean	nominal	'good'	As VWO_mean
HO_Education	nominal	'Electrical'	{n/a, Electrical, Technical, Other}
HO_Year	nominal	2001-2003	Same categories as VWO_Year
HO_Grade	nominal	'good'	As VWO_mean
Gapyear	nominal	-1	{n/a, <-1, -1, 0, 1, >1 }
Classification	nominal	1	{-1, 1}

Table 2: Attributes of past education dataset

to the VWO grades, in this case the grades are stored as numerical attributes instead of nominal attributes. This is done because we expect more variability in the grade distribution among the different courses, making it less trivial how to transform the data. Furthermore, we do not want to lose or misrepresent information using a suboptimal discretization. Therefore, the data is kept as nominal data.

Using old study guides, the university website and other sources, the course names belonging to the given course codes were retrieved. The result is listed in table 3. Having this information, a dataset could be composed.

In these preparatory steps it became clear that the stored data is not very homogeneous for all students. There are different types of examinations in different years, some courses were moved in time and others were combined. Nevertheless it makes sense to build a model based on this in homogeneous dataset, because while the curriculum *organization* was changing frequently, the curriculum *contents* have not changed considerably since 2001.

Course	Types	Details
Calc_	A, AB, B, 1, 2, 3, 4, 5	Calculus course
Comp_	A, A1, A2, AB, B	Computation course
LinAlg_	A, AB, B, 1, 2, 3, 4, 5	Linear algebra course
Netw_	A, A1, A2, AB, B, B1, 1, 2, 3, 4, 5	Networks course
Project	-	Project work
TIO	-	Theory Integrating Assignments Lab
OtherEle	-	All electrical engineering courses not considered above
OtherMath	-	All mathematics courses not considered above
OtherX	-	All other courses

Table 3: Courses and course types in grades dataset

5.3 Classification

Besides preparing these two data sets, a classifier is needed to be able to learn a model. As became clear above, the dataset seems not very rich, some attributes will be strongly intercorrelated and the population is quite heterogeneous due to changing curriculum organization. This will not make it easier to get a very sophisticated and precise classifier. Therefore, as first attempt it is chosen to pick a binary classifier. The students are classified in the following way: if a student was able to get his 'propedeuse' — that is, to complete all courses of the first year — in three years, he is classified as successful, if he did not, he is classified as unsuccessful. Obviously, this is a very rude and clear criterion. The drawback is that we can not use some of the student data of the last three years. These students are therefore left out of the dataset. This leaves us with 516 records in both datasets to learn a model.

For practical use of the resulting model, it would be much more convenient if the classification could be such that we get three classes: bad, risk, good, so resources can be targeted mainly on the risk group. Furthermore, it would be very useful if the classification did not exclude the more recent students. Therefore, in accordance with a rule of thumb used in practice, as second classification we use: Students obtaining less than 50% of the nominal amount of credits after one year are classified as bad, students obtaining 50-80% are classified as risk, and students obtaining more credits are classified as good. Note that using this classifier, we are able to include students of more recent years which have followed other sub-courses, which might influence the dataset significantly.

To summarize the properties of the final datasets used in these preliminary steps the most important facts are shown in table 4. The increased number of students of recent years is clearly visible.

6 Preliminary mining

Preliminary mining is done to investigate mainly three aspects. Firstly different data mining techniques are studied to see which are the most suitable for this dataset. As already stated in section 2, decision trees are slightly preferred in this field. However, the evidence is not conclusive. Therefore, several techniques are investigated to see if there are significant differences. Secondly, the two different sections of the dataset are compared with each other. And thirdly, the main predictors

	Dataset1	Dataset2
number of students	516	549
Classification	(-1,1)	(bad,risk,good)
Class distribution	(253,263)	(189,82,278)
Total VWO	468	507
<2001	121	120
2001-2003	136	127
2004-2006	204	201
2007-2008	7	59
Total HO	58	50
HO and VWO	31	29
No data	21	21

Table 4: Summary of the two used datasets

of the data are determined. In this preliminary research, only the dataset with the binary classifier is used.

We compare two decision tree algorithms (`SimpleCart` and `J48`), a Bayesian classifier (`BayesNet`), a logistic model (`SimpleLogistic`) and a rule learner (`JRip`). The minimum-error attribute selector `OneR`is used as base model. To get an indication of the achievable performance, a Random Forest algorithm is added too (`RandomForest`). It is set to generate 100 random trees (default 10). The `J48` algorithm is run twice: once with the default options, and once with a minimum number of instances per leave of 10 (default is 2). This is done to limit the size of the tree. All the other learners are run with default settings.

6.1 Description of the data mining techniques employed

Before the conducted experiments are treated, a short introduction to the algorithms mentioned above will follow here. For unexplained terminology and more details, see for instance [15], [16] and [17].

Decision tree learners are algorithms that construct a decision tree as classifier. A decision tree is a hierarchical structure, consisting of nodes and directed edges. Its base is the *root node*, which has no incoming edges and has some outgoing edges. Every outgoing edge points to an *internal node* or to a *leaf node*. An internal node has one incoming edge and two or more outgoing edges, while a leaf node has one incoming edge and no outgoing edges. The root node and the internal nodes all contain a test condition, used to separate records. For instance, at a certain node the attribute *VWO_mean* might be used to separate between good and less good VWO students. After the separation, the corresponding edge is taken and one arrives either at another internal node (where this procedure is repeated for another attribute) or at a leaf node. With every leaf node, a class label is associated. In this way, a decision tree is able to classify all records.
In building up a tree, the algorithm seeks to create a set of leaf nodes that is as "good" as possible: in every leaf node it seeks to have only records belonging to one particular class. In this way, the classifier achieves high accuracy.
Tree algorithms differ mainly from each other in the way how "goodness" or "purity" of a given

leaf is measured. The two main methods used are Gini impurity and entropy gain, where the first is used by the CART algorithm and the latter in the C4.5 and C5.0 algorithms. `SimpleCART` is an implementation of the CART algorithm, `J48` is an implementation of the C4.5 algorithm.

Bayesian classifiers try to model probabilistic relationships between the attributes and the class variable. It uses the well-known *Bayes theorem* to combine *a priori* information with evidence from data. Let X denote the attribute set and Y denote the class variable. The classifier learns the *posterior probablity* $P(Y|X)$ for every combination of X and Y, so a new record can be classified such that the posterior probability is maximal. This technique is used in different ways for different algorithms, where `BayesNet` is an implementation of a *Bayesian network*.

Logistic models use *linear logistic regression* known from statistics to calculate the probability of the class variable given the record under study. The Weka implementation is called `SimpleLogistic`.

Rule-based learners are algorithms that try to find a set of rules used for classification, in the form "if ... then class = ... ". Rules might or might not be created as mutually exclusive set. If the rule set is not mutually exclusive, the potential conflicts resulting from this must be solved. Algorithms differ in which method is used to evaluate the quality of a potential rule. The widely used rule induction algorithm RIPPER is implemented in Weka as `JRip`.

Random forest is an *ensemble method*, which means that it aggregates the prediction of multiple classifiers to improve accuracy. The random forest algorithm selects a subset of the input features as training set, and uses decision trees as its base classifier. The resulting decision trees are combined to get a good accuracy. The gain compared to single decision trees can sometimes be impressive, but sometimes there is no improved accuracy at all (see [17, p.294] for an empirical comparison).

6.2 Mining on pre-university data

These classifiers are run on the dataset containing the pre-university data. We use 10-fold cross validation, because the dataset is too small to split into test data and validation data. The predictive capabilities of the models are compared and tested on significance using `Weka Experimenter`, which uses a corrected paired T-Test for this. The base model took the average science grade as selector, and achieved an accuracy of 68%. Running the other algorithms showed that none of them was able to raise the accuracy significantly. Even the random forest algorithm was not able to produce a significantly better result. See the results in table 5

Algorithms	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Accuracy	68.2	67.6	69.9	69.2	71.1	68.8	69.6	65.3

(1) rules.OneR '-B 6'
(2) trees.SimpleCart '-S 1 -M 2.0 -N 5 -C 1.0'
(3) trees.J48 '-C 0.25 -M 2'
(4) trees.J48 '-C 0.25 -M 10'
(5) bayes.BayesNet '-D -Q ... K2 – -P 1 -S BAYES -E ... SimpleEst. – -A 0.5'
(6) functions.SimpleLogistic '-I 0 -M 500 -H 50 -W 0.0'
(7) rules.JRip '-F 3 -N 2.0 -O 2 -S 1'
(8) trees.RandomForest '-I 100 -K 0 -S 1'

Table 5: Accuracy pre-university data

To explain this, the attributes are ranked using the information gain evaluator of `Weka`. This ranks the attributes on their information gain (using an entropy measure) with respect to the class. This showed that the attributes *VWO_Science_mean*, *VWO_main* and *VWO_Math_mean* were by far the best attributes in information gain (information gains 0.16, 0.13, 0.12 respectively), with the next attribute *VWO_Year* lagging behind (gain of 0.05). Furthermore, these three attributes are highly correlated. Therefore, there is not much information in the set to learn more than the base model already did. This explanation is confirmed by learning a `J48` tree using only the three mentioned attributes, which achieves an accuracy of 71%.

Although this means that the amount of information in this dataset is not very high, it does not mean that the dataset is useless. It might be that some attributes are strong predictors in later leaves of a tree that is built using university grade attributes. It remains to be seen whether this is so indeed.

6.3 Mining on university grades

The same test is run on the university grades, the results are shown in table 6. The base model selects the grade for Linear Algebra (*LinAlgAB*), and decides positive if this grade is bigger than 5.5. This is a satisfying result: a grade of 5.5 is exactly the minimum for passing a course. Again, we see that data mining algorithms are not able to improve accuracy very much. However, now one algorithm is significantly better than the base line model. The CART tree builder succeeds in outperforming the base line model. The other models were not significantly better or worse than the base line model.

The CART model produced uses *LinAlgAB* as root of the tree, and uses *CalcA*, *Calc1* and *Project_nAttempts* as further discriminators, in a relatively small tree with five leaves. It is remarkable that the grades of the Networks course are not used at all, while some of it's related attributes have a high information gain measure (ranking 4,5 and 6, just after three Linear Algebra related attributes). Checking the correlation plots however does show some correlation between Linear Algebra and Networks attributes, while there is less correlation observed between Linear Algebra and Calculus attributes. This might be the reason of this effect.

Algorithms	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Accuracy	76.0	80.8 ∘	79.2	78.7	75.1	79.4	77.6	80.4

∘ statistically significant improvement

(1) rules.OneR '-B 6'
(2) trees.SimpleCart '-S 1 -M 2.0 -N 5 -C 1.0'
(3) trees.J48 '-C 0.25 -M 2'
(4) trees.J48 '-C 0.25 -M 10'
(5) bayes.BayesNet '-D -Q ... K2 – -P 1 -S BAYES -E ... SimpleEst. – -A 0.5'
(6) functions.SimpleLogistic '-I 0 -M 500 -H 50 -W 0.0'
(7) rules.JRip '-F 3 -N 2.0 -O 2 -S 1'
(8) trees.RandomForest '-I 100 -K 0 -S 1'

Table 6: Accuracy of university grades

6.4 Mining on the total dataset

Finally, the total dataset is investigated in the same manner as the subdatasets were before. The results are shown in table 7. The accuracy is comparable with the accuracy in the previous subdataset. Apparently, the pre-university data does not add much information that can be improve classification accuracy. However, we can see that the trees learnt using `J48` are now statistically significantly better than the baseline model. The other tree algorithms are also doing well, while the Bayes classifier and the rule learning algorithm slightly fall behind.

To get a better insight on the performance of the classifiers, the scoring of the algorithms is shown in more detail now. A remarkable fact is that the base line model has a higher false negative rate than all other models. This is an interesting finding, because giving a negative advice incorrectly might be considered more severe than giving a positive advice incorrectly. Data mining techniques can be used to manipulate this distribution of errors.

Algorithms	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Accuracy	74.6	79.3	79.9 ∘	79.5	75.4	79.3	77.4	79.4
True positive rates	0.64	0.79 ∘	0.80 ∘	0.75 ∘	0.72 ∘	0.79 ∘	0.73 ∘	0.82 ∘
False negative rates	0.36	0.21 ∘	0.20 ∘	0.25 ∘	0.28 ∘	0.21 ∘	0.27 ∘	0.18 ∘
True negative rates	0.86	0.80 •	0.80 •	0.84	0.79 •	0.80 •	0.82	0.77 •
False positive rates	0.14	0.20 •	0.20 •	0.16	0.21 •	0.20 •	0.18	0.23 •

∘, • statistically significant improvement or degradation

(1) rules.OneR '-B 6'
(2) trees.SimpleCart '-S 1 -M 2.0 -N 5 -C 1.0'
(3) trees.J48 '-C 0.25 -M 2'
(4) trees.J48 '-C 0.25 -M 10'
(5) bayes.BayesNet '-D -Q ... K2 – -P 1 -S BAYES -E ... SimpleEst. – -A 0.5'
(6) functions.SimpleLogistic '-I 0 -M 500 -H 50 -W 0.0'
(7) rules.JRip '-F 3 -N 2.0 -O 2 -S 1'
(8) trees.RandomForest '-I 100 -K 0 -S 1'

Table 7: Accuracy and rates of total dataset

6.5 Conclusions on preliminary mining

These preliminary mining results can be summarized in the following considerations.

It proves to be difficult for all models to raise above the predicting power of the base line models. Apparently, the dataset is inherently difficult to model. However, the decision tree algorithms seems to be slightly better than the other algorithms: the CART and C4.5 algorithms were the only ones capable of improving the base line model significantly. Therefore, these are the algorithms to be used in the remainder of the paper.

When we compare the different sections of the dataset, it is clear that university grades are a better predictor than pre-university data. The pre-university data attributes seem to be either irrelevant or strongly correlated, resulting in a prediction accuracy of around 70% using the mean grade of the science courses as predictor. The university grades have a slightly more complex structure, shown by the fact that it is possible here to rise above the base line model. Comparing the different resulting models, the attribute *LinAlgAB* is clearly the dominant predictor, followed by Calculus related attributes. This is also the case in the models of the total dataset. However, sometimes an attribute of the pre-university dataset pops up. In the remainder of the paper only the combined dataset is used, because this section showed that considering both subdatasets separately does not appear to be useful.

Using the full dataset, the accuracy of the prediction cannot be improved easily using data mining. However, as shown in table 7, it might be able to shape the models such that different types of errors or classifications are preferred or rejected. This is the topic of the next section.

7 Using classification weights

To determine how the different errors should be penalized, the student counselor was consulted. He indicated that it is better to give an erroneous positive advice to a student who should actually be classified as negative, than to give a erroneous negative advice to a student who should be classified as positive. A negative advice can work demotivating on a student who is just able to be successful, and can thereby work as a self-fulfilling prophecy.

7.1 Using cost matrices

To be able to incorporate these different misclassifications, we should have a look at the confusion matrices produced by the learning algorithms applied on the dataset. It shows how data instances are classified. A confusion matrix produced by `Weka` based on the (-1,1) classification has the following structure: where TN means true negative, FN means false negative and so on. The

	classified as negative	classified as positive
actual negative	TN	FP
actual positive	FN	TP

accuracy can be calculated from this matrix using $\frac{TN+TP}{TP+TN+FP+FN}$. The preference we have can now be stated as follows: we prefer the algorithm to produce a FP misclassification to a FN misclassification.

To be able to take this preference into account, cost matrices can be used. A cost matrix encodes the penalty of classifying records from one class as another. The used cost matrix is represented in the following structure, in accordance with the confusion matrix :

	classified as negative	classified as positive
actual negative	$C(-,-)$	$C(-,+)$
actual positive	$C(+,-)$	$C(+,+)$

By choosing the weights $C(i,j)$ appropriately, preferences in classifications can be taken into account. The overall cost of a model M is

$$C(M) = TP * C(+,+) + FP * C(-,+) + FN * C(+,-) + TN * C(-,-) \quad (1)$$

Therefore, using the diagonal entries is especially useful if there is class imbalance, while preferring certain misclassifications above others can be achieved using the off-diagonal entries. Cost matrices are equivalent under scaling, and because we only want to weigh the preference of false positives over false negatives, it suffices to build a matrix with only one free coefficient and structure $\begin{bmatrix} 0 & 1 \\ C(+,-) & 0 \end{bmatrix}$. We want to increase the cost of false negatives over false positives, so we take $C(+,-) > 1$. The final weight will be determined during the experiment.

7.2 Experiment setup

In this experiment the metalearner `CostSensitiveClassifier` of `Weka` is used. It takes the given cost matrix into account in evaluating its base classifier. It supports two ways to do this. The first way is to use equation 1 directly in choosing the best model. This method is called *Model cost* in table 8. The second way is to use the cost matrix as a bias for the training data: the training data is then reweighted according to the costs assigned to each class. This is called *Data weighting* in the table. These two ways of using the weights might result in slightly different outcomes, so they are both used.

The earlier examined `J48` and `CART` are used as base classifiers. In addition, some experiments are carried out using `J48graft`, an improved version of `J48`. This algorithm is however likely to grow into very complex models [18]. To prevent the tree from growing too big, a preselection in the attributes is made using the `CfsSubsetEval`. This algorithm tries to select a small subset of attributes that have low intercorrelation, while having a high correlation with the class. Furthermore, while using the `J48` as well as the `J48graft` algorithm, the minimum number of instances of a leave was set to 10 (standard setting 2). In this way, overfitting and unnecessarily complex models could be prevented.

Combining these three algorithms with the two ways of using the cost matrix, six experiments are conducted. The different outcomes have to be evaluated. To be able to do this, the recall and precision metrics are used. Recall r is defined as $r = \frac{TP}{TP+FN}$. A classifier with a large recall will have very few positive examples misclassified as the negative class. This is the major objective in this setting. The precision metric p is defined as $p = \frac{TP}{TP+FP}$. This defines the fraction of records that actually is positive in the group that is classified as positive. If all effort would be in increasing r, the classifier would choose to label all instances as positive. However, this would decrease p. Therefore, a trade-off between r and p should be found. This trade-off can be examined defining a metric known as the F_β measure, which is defined as

$$F_\beta = \frac{\left(\beta^2 + 1\right) rp}{r + \beta^2 p} \tag{2}$$

Choosing $\beta = 1$ would give the harmonic mean of r and p, while setting $\beta > 1$ would give more weight to recall. Recall is more important to us than precision, and after some experimenting β is set to 1.5.

Using these performance measures, the cost matrix parameter $C(+,-)$ is varied for all six combinations of algorithm settings. For each combination, the settings giving the highest F_β measure are selected and displayed in table 8. As first entry a `J48` model is used without cost matrix.

	1	2	3	4	5	6	7	8
Type	J48	J48	J48	CART	CART	CART	J48graft	J48graft
Learner option	-	Data weighting	Model cost	Data weighting	Model cost	Model cost	Data weighting	Model cost
$C(+,-)$	-	2	3	2	3	4	4	3.2
Confusion Matrix	212 41 65 198	175 78 49 214	206 47 62 201	169 84 50 213	201 52 57 206	181 72 51 212	160 93 31 232	161 92 56 207
Accuracy	0.79	0.75	0.79	0.74	0.79	0.76	0.76	0.71
Precision	0.83	0.73	0.81	0.72	0.80	0.75	0.71	0.69
Recall	0.75	0.81	0.76	0.81	0.78	0.81	0.88	0.79
F_β	0.77	0.79	0.78	0.78	0.79	0.79	0.82	0.76
nLeaves	5	11	5	10	7	7	21	8
TreeDepth	3	6	3	5	4	4	8	5
Root node	LinAlgAB <= 5	LinAlgAB <= 5	LinAlgAB <= 5	LinAlgAB <5.5	LinAlgAB <= 5.5	LinAlgAB <= 5.5	LinAlgAB <= 5	LinAlgAB <= 5
First node	NetwB <= 5.7	CalcA <= 5	NetwB <= 5.7	VWO-Science-mean	CalcA <5.15	CalcA <5.15	CalcA <= 5	CalcA <= 5
Second node	CompB-nAttempts	CompB-nAttempts	CompB-nAttempts	LinAlgA, CalcA	VWO-Science-mean	VWO-Science-mean	VWO-Science-mean	LinAlgB, NetwA2

Table 8: Cost matrix results

7.3 Evaluation of results

The results show that it is necessary to sacrifice some of the achieved accuracy to be able to shape the misclassification: only model 5 achieves a high accuracy and a high F_β measure, all other models lose in accuracy if F_β is increased. During the experiment, it became clear that there is not much room for enhancement: if r was increased to values higher than 85%, this led to unacceptable accuracy results, with model 7 as only exception. In some cases, small trade-offs could be made changing C. Compare for instance model 5 with model 6: a three percent point drop in accuracy gives a three percent rise in recall. However, the changes is the model are only minor: the top part of the tree is identical.

The created decision trees are remarkably similar: in every tree the *LinAlgAB* attribute is dominant, with *CalcA* as first node in most of the cases. When *NetwB* is chosen as first node, recall is lower, although the difference is too small to draw decisive conclusions.

Taking the accuracy and the F_β measure as guide, the models 5, 6 and 7 perform best. The difference with the base line model is not very big, except in the case of model 7. However, model 7 is a complex tree which might be too detailed to be meaningful.

We conclude that is seems to be difficult to shape the misclassification costs significantly, although small improvements can be made. The trees are remarkably similar which gives confidence in the stability of the models. However, new directions are needed to improve the model.

8 Model evaluation

As final step, one of the models is examined into detail to see if we can gain understanding about the classifier errors. The student counselor compared all the wrongly classified instances of model 7 with his own given advices to check for interesting patterns.

One of the first things that is assessed is the question whether the learned model is incorrect or the classification measure is chosen incorrect. To examine this, two methods are used. Firstly, the false negative and false positive sets have been checked manually by the student counselor. His conclusions are that about 25% of the false negatives should be true negatives instead. This might indicate a wrong classification measure. Concerning the false positive set a conclusion is less obvious: about 45% of this set was classified as positive by the study counselor as well as by the tree, but did not meet the classification criterion. a substantial subset of these students have chosen not to continue their bachelor program in Electrical Engineering although all indications for a successful continuation were present. Qualifying these students as "false" positive does not seem to be appropriate. So from this evaluation based on domain expertise we can conclude that some of the mistakes might be due to the classification measure, and some of them raise suspicion on behalf of the learned model.

The second way to check the viability of the model is to compare the results obtained with this classifier with those of the classifier of dataset 2 (see table 4). If the learned model would be performing good, it is expected that the wrongly classified students will be in the "risk" classification of dataset 2. In that case, the model has difficulties in predicting the students who are difficult to classify in success or failure categories indeed, because they are on the boundary of these classes. However, we observe that only 25% of the misclassified instances are in this category. It should be noted that this is still twice as much as the risk students ratio in the total dataset. Therefore, this also indicates that the learned model should be improved. Furthermore, 25% of the instances in the false positive class would be classified as "good" using the three-class classification. This indicates a real difference between both classifiers. So from this test we can also conclude that the model as well as the classification measure need to be reviewed to increase performance.

After these tests, the misclassified sets are looked up in the database to search for meaningful patterns manually. A very clear pattern popped up immediately: almost all misclassified students did not have a database entry concerning *LinAlgAB*. Therefore, their *LinAlgAB* entries were mapped to zero during the data transformation into `arff`-format. Checking out different students showed that there are many possible reasons now to have a zero value in the *LinAlgAB* record:

* A student might be of a cohort in which the *LinAlgAB* exam was in January or later;
* A student might have not shown up during the exam;
* A student might have taken another way to get its *LinAlgAB* grade: in some years it was possible to bypass the regular exam by doing the subexams *LinAlg1*, *LinAlg2*, *LinAlg3*, *LinAlg4* and *LinAlg5*. A student succeeding in taking this path can well be an excellent student, but get a zero mark for the *LinAlgAB* attribute.

Due to this effect, 216 of the 516 students do have a zero entry in their *LinAlgAB* record. Moreover, the same effect will play a role for the other courses too.

9 Discussion

At the end of this report the interests of the department in this topic are revisited. Starting with the question which attributes are the strongest predictors of success, the conclusion is warranted that the *LinAlgAB* attribute is the strongest predictor available. According to the student counselor, this is a surprising result: the linear algebra course is in general not seen as the decisive course. Other strong predictors are *CalcA*, *NetwB* and *VWOScienceMean*. The most relevant information is collected at the university itself at the moment: the pre-university data can be summarized into one attribute.

Assessing the performance of the predictor, we can conclude that it is difficult to learn a complex model using the current dataset. Very simple classifiers give a useful result with accuracies between 75 and 80%. It proved to be possible to take a relative preference of false positives to false negatives into account using cost matrices. In this approach, the gain was not very big but it was certainly there.

The model evaluation points to three major improvements that can be assessed. Firstly, a key improvement in this dataset is to find a solution for the changing course organization over the set. Aggregating the available information about student performance for a course in a way that can be used for all subjects in the dataset might prevent the type of misclassifications that is now strongly prevalent. Some of the preprocessing steps taken should be revised and improved to achieve this.

A second, related improvement would be a better way to encode grades in general. Mapping all unknown or not available information to zero showed to be not effective. Specifically, Linear Algebra grades should be available. A more advanced solution dealing with missing values also can be considered in this respect. In this paper we experimented with the so-called 0/1 loss and cost-sensitive classification. AUC optimization is also one of the directions of further work.

The quality of the classification measure is the third improvement that might be considered. The binary classification as used in section 5.3 is simple, but has its difficulties too: a negative classification can only be given after three years, and there is no guarantee that a student who does not get his propedeuse after three years will be not successful in the long run. Also, students who do not receive a propedeutical diploma should not necessarily be disqualified: they may have had different motives to discontinue their studies. This touches on a more fundamental topic: to objectively classify "good" students needs more careful consideration. As a first improvement, the three-class classification measure as used in this study might be employed. In that way, it would also become possible to classify students as belonging to the "risk" class.

This study is not conclusive on what data might result in a further increase of the predictor quality. To be able to determine the necessity of additional data collection, the improvements mentioned above should be implemented first to be able to determine whether additional data collection makes sense. However, the literature review showed that sociological and demographical data can be predictors in some cases. Therefore it can be expected that the use of these kind of data can improve the model. In the situation of this case study, I would say that the residential situation of a student (independently or with his parents) and his travel time might be important. Whether a student is building a social network at the department (and therefore having more to lose) might be important too. If this information can be made available easily, it seems worthwhile to investigate its use.

Finally, this study shows that data mining techniques can also have their use on less rich datasets, provided the data preparatory steps are carried out carefully. The basic analysis presented here shows an accuracy of 75% to 80% based on pre-university data and first-semester data , and shows several ways of possible improvement without having to collect additional data.

References

[1] G. W. Dekker, M. Pechenizkiy, and J. M. Vleeshouwers, "Predicting students drop out: A case study." Submitted to the 2nd Int. Conf. on Educational Data Mining (EDM '09), 2009.

[2] C. Romero and S. Ventura, "Educational data mining: a survey from 1995 to 2005," *Expert Systems with Applications*, no. 33, pp. 135–146, 2007.

[3] J. Superby, J.-P. Vandamme, and N. Meskens, "Determination of factors influencing the achievement of the first-year university students using data mining methods," in *Proceedings of the Workshop on Educational Data Mining at the 8th International Conference on Intelligent Tutoring Systems (ITS)*, pp. 37–44, 2006.

[4] V. Tinto, "Limits of theory and practice in student attrition," *Journal of Higher Education*, no. 53, pp. 687–700, 1982.

[5] P. T. Terenzini, W. G. Lorang, and E. T. Pascarella, "Predicting freshman persistence and voluntary dropout decisions: a replication," *Research in Higher Education*, vol. 15, no. 2, pp. 109–127, 1981.

[6] P. Parmentier, *La réussite des études universitaires: facteurs structurels et processuels de la performance académique en première année en médecine.* PhD thesis, Faculté de Psychologie et des Sciences de l'Éducation, Catholic University of Louvain, 1994.

[7] J. Touron, "The determination of factors related to academic achievement in the university: implications for the selection and counselling of students," *Higher Education*, vol. 12, pp. 399–410, 1983.

[8] G. Lassibille and L. N. Gómez, "Why do higher education students drop out? Evidence from Spain," *Education Economics*, vol. 16, no. 1, pp. 89–105, 2007.

[9] S. Herzog, "Measuring determinants of student return vs. dropout/stopout vs. transfer: A first-to-second year analysis of new freshmen." Online Submission, Paper presented at the 44th Annual Forum of the Association for Institutional Research (AIR), 2004.

[10] S. Herzog, "Estimating student retention and degree-completion time: Decision trees and neural networks vis-à-vis regression.," *New Directions For Institutional Research*, pp. 17–33, Fall 2006.

[11] J. Luan, "Data mining and its applications in higher education.," *New Directions For Institutional Research*, pp. 17–36, Spring 2002.

[12] N. Thai Nge, P. Janecek, and P. Haddawy, "A comparative analysis of techniques for predicting academic performance," in *37th ASEE/IEEE Frontiers in Education Conference*, 2007.

[13] C. Romero, S. Ventura, P. G. Espejo, and C. Hervás, "Data mining algorithms to classify students," in *Proceedings of the first International Conference on Educational Data Mining* (R. S. J. de Baker, T. Barnes, and J. E. Beck, eds.), pp. 8–17, 2008.

[14] M. Pechenizkiy, T. Calders, E. Vasilyeva, and P. de Bra, "Mining the student assessment data: Lessons drawn from a small scale case study," in *Proceedings of the first International Conference on Educational Data Mining* (R. S. J. de Baker, T. Barnes, and J. E. Beck, eds.), pp. 187–191, 2008.

[15] D. Larose, *Discovering knowledge in data: an introduction to data mining.* Hoboken: John Wiley & Sons, Inc., 2005.

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

[17] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining.* Boston: Pearson Addison-Wesley, 2006.

[18] G. I. Webb, "Decision tree grafting," in *IJCAI-97: Fifteenth International Joint Conference on Artificial Intelligence*, pp. 846–851, Morgan Kaufmann, 1997.