# 1 Introduction

With the rise of technology-enhanced education, in particular when software-based, large and diverse groups of students are interacting with the learning materials (REF(s)). These interactions usually take place in an online environment where the role of teachers in tutoring and assessment is limited compared to the traditional classroom setup. Therefore, intensive support for increasing students' learning effectiveness and feedback to students from an educator is of limited availability. When these online learning environments are augmented with techniques keeping track of students' characteristics and interactions with the software, new possibilities open up for feedback for students and teachers. Being able to identify personal learning styles, goals and indications for student progress is therefore an integral part of data-driven approaches used in intelligent learning environments (ILE's) for building student models. Student characteristics can be obtained through interviews or as a form that is part of the ILE. Features such as age, motivation, pre-knowledge may be indicative features for an outcome variable such as learning effectiveness or the possession of certain related knowledge. The interactions with the software can be logged and exported and be used for the same purpose. In this paper for example, we want to check which features and interactions are useful predictors, and in what way, for the end result on the exam of the students using our software. These predictions can be useful as feedback to either the instructor or the student. A teacher might for example choose to give students with a low end-grade prediction extra attention. Students themselves might get more motivated to intensify their learning due to knowledge of their predicted grade. Our application will include an 'analytics dashboard' that can be accessed by either the instructor or the student (with an individual view-mode for the student), which should present clear and incisive graphical feedback on the aforementioned aspect of end result predictions.

Machine Learning, with its ability to recognize patterns from data, is a natural candidate for tasks dealing with learning from student's interactions in an ILE. It is shown by [6] that historically, machine learning has been mostly used in two important areas of research in student modeling: automatically extending or constructing from scratch the bug library of student modelers as in PIXIE [8] and MEDD [7] and for building a consistent and insightful student model by induction from student interaction data such as in DEBUGGY [1] and THEMIS [5]. More recently, machine learning techniques have been combined with the approach of stereotyping to cluster all possible users of an adaptive system into several groups according to certain characteristics that they typically share [3]. In [2] Dynamic Bayesian Networks were used to select the appropriate learning resources for each student alongside stereotyping to classify students based on their learning style. In [9] stereotypes are used to classify a new learner based on initial assumptions. The student is then compared with the k-means algorithm to students in the same group to estimate the knowledge the learner already has.

Although the approach of stereotypes as a student model has matured and many effective and interesting possibilities have been explored, some difficulties remain unsolved. In a literature review paper about student modeling in general [3], the authors note two problems with stereotyping:

> First, in order to use them, the set of system users must be divisible into classes; however, such classes may not exist. Second, even if it is possible to identify classes of system users, the system designer must build the stereotypes; this is a process that is both time-consuming and error-prone. [4]

Furthermore, [9] focusses on the effect of manually creating stereotypes:

> The stereotype approach is quite inflexible due to the fact that stereotypes are constructed in a hand-crafted way before real users have interacted with the system and they are not updated until a human does so explicitly. [9]

In this paper we introduce Machine Learning Stereotypes (MLST), which utilizes 'decision trees' machine learning as an underlying technique for constructing and evaluating stereotypes. Decision trees can partition the students in clusters (stereotypes) based by repeatedly partitioning the n-dimensional feature space into two parts that have a meaningful difference with respect to the result that we are interested in (in our case the end result on the exam). The method of decision trees is not always the most optimal algorithm with respect to performance in prediction success. An important advantage for our work however, is that the partitioning mechanism itself produces meaningful semantic information, as it splits on a specific feature (for example age), which produces an intuitive term (e.g.: 'older than 20') for the stereotype. Since we develop this algorithm and its graphical extension (the analytics dashboard) to provide feedback to users and instructors, this semantic information is an important feature. Intuitive labels, being either a single feature split term, or an aggregation of split terms provide teachers with feedback that they can actually work with, instead of abstract or complex numerical machine learning information. The algorithm itself may be convoluted (e.g.: interplay between variables), the presentation to the user should not.

We use the software 'BattleQuiz' to develop our algorithm and apply it to the learners of the system. 'BattleQuiz' is a gaming platform that enables users to compete against the computer or peers in multiple-choice quizzes. As it is a platform, the content is not fixed, it is based on the learning goals of the institution that utilizes it. In our case, MLST will be applied to the content of 'VAC-bouwveiligheid', a Dutch course in construction safety.

In the next section we will discuss work that is related to our research. Next, we will present our stereotype design MLST, based on the technique of 'decision trees' and how we implement this in the online gaming platform 'BattleQuiz'.

Then, we will explain our method of evaluating and interpret our results. We conclude with suggestions for future work and summarize our approach.

## References

[1] Richard R Burton. Diagnosing bugs in a simple procedural skill. In *Intelligent tutoring systems*. Academic press, 1982.

[2] Cristina Carmona, Gladys Castillo, and Eva Millán. Designing a dynamic bayesian network for modeling students' learning styles. In *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on*, pages 346–350. IEEE, 2008.

[3] Konstantina Chrysafiadi and Maria Virvou. Student modeling approaches: A literature review for the last decade. *Expert Systems with Applications*, 40(11):4715–4729, 2013.

[4] Robert Kass. Building a user model implicitly from a cooperative advisory dialog. *User Modeling and User-Adapted Interaction*, 1(3):203–258, 1991.

[5] Yasuyuki Kono, Mitsuru Ikeda, and Riichiro Mizoguchi. Themis: a non-monotonic inductive student modeling system. *Journal of artificial Intelligence in Education*, 5:371–371, 1994.

[6] Raymund Sison and Masamichi Shimura. Student modeling and machine learning. *International Journal of Artificial Intelligence in Education (IJAIED)*, 9:128–158, 1998.

[7] Raymund Sison, Masayuki Numao, and Masamichi Shimura. Discovering error classes from discrepancies in novice behaviors via multistrategy conceptual clustering. *User Modeling and User-Adapted Interaction*, 8(1-2): 103–129, 1998.

[8] Derek Sleeman. Pixie: A shell for developing intelligent tutoring systems. In *Artificial Intelligence in Education*, volume 1, pages 239–265, 1987.

[9] Victoria Tsiriga and Maria Virvou. Initializing the student model using stereotypes and machine learning. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*. Citeseer, 2002.