Demonstration of a Moodle student monitoring web application

Angela Bovo
Andil
Université Toulouse 1
IRIT
angela.bovo@andil.fr

Stéphane Sanchez
Université Toulouse 1
IRIT
sanchez@univ-tlse1.fr

Olivier Héguy
Andil
olivier.heguy@andil.fr

Yves Duthen
Université Toulouse 1
IRIT
yves.duthen@univ-tlse1.fr

ABSTRACT

We would like to demonstrate a web application using data mining and machine learning techniques to monitor students's progress along their e-learning cursus and keep them from falling behind their peers.

Keywords

Moodle, analysis, monitoring, machine learning, statistics

1. AIMS OF THE APPLICATION

We would like to demonstrate a web application developed during a project supported jointly by computer science researchers and an IT firm specialized in e-learning software. Another partner firm, a professional training institute, connects our project with real data from its past and current e-learning courses on various Moodle platforms.

The aim of our application is to use the methods of IT, data mining and machine learning to give educators better tools to help their e-learning students. More specifically, we want to improve the monitoring of students, to automate some of the educators' work, to consolidate all of the data generated by a training, and to examine this data with classical machine learning algorithms. This application is called GIGA, which means Gestionnaire d'indice général d'apprentissage (French for General Learning Index Manager).

The reasons for monitoring students are that we want to keep them from falling behind their peers and giving up, which can be noticed earlier and automatically by data mining methods; we also want to see if we are able to predict their end results at their exams just from their curriculum data, which would mean we could henceforth advise students on how they are doing.

However, currently, Moodle offers very few statistics, and they are hard to examine and analyse. Notably, they are purely individual, so we cannot have a global vision of a group or compare students. We can view a list of logs but hardly anything synthetic, except a few graphs for login data. Only the date of last login and the number of quizzes done were actually used by the training manager we talked to, which seems a waste compared to all the logging done by Moodle. Hence, the need felt for our application.

2. DESCRIPTION OF THE CURRENT IMPLEMENTATION

The web application that we wish to demonstrate is already in use for student monitoring in our partner training institute. This application gathers data from a LMS and other sources and allows to monitor students with raw figures, statistics and machine learning.

Our implementation uses the language Java with frameworks Wicket, Hibernate, Spring and Shiro. The data is stored in a MySQL database.

In our case, the LMS is a Moodle platform where the courses are located. Moodle registers some events in its logging system, which we then import and mine. Hence, we are also constrained by what Moodle does and does not log. For instance, our partner firm had to create a Moodle plugin to have better logout estimates, that will be deployed on future trainings. However, the application could be very simply extended to other LMSes that have a similar logging system.

2.1 Data consolidation

We have decided to consolidate into a single database most of the data produced by an e-learning training. Currently, the data is scattered in two main sources: the students' activity data are stored by the LMS, whereas some other data (administrative, on-site training, contact and communication history, final exams grades) are kept by the training managers, their administrative team and the diverse educators, sometimes in ill-adapted solutions such as in a spreadsheet. This keeps teachers from making meaningful links: for instance, the student has not logged in this week, but it is actually normal because they called to say they were ill.

We have already provided forms for importing grades obtained in offline exams, presence at on-site trainings and commentaries on students. In the future, we will expand this to an import directly from a spreadsheet, and to other types of data. From Moodle, we regularly import the relevant data: categories, sections, lessons, resources, activities, logs and grades.

2.2 Data granularity

All raw data imported from Moodle or from other sources is directly available for consultation, such as the dates and times of login and logout of each student, or each grade obtained in quizzes.

We then provide statistics built from these raw data, such as the mean number of logins over the selected time period. This is already a level of granularity not provided by Moodle except in rare cases.

We also felt a need for a normalized indicator that would make our statistics easy to understand, like a grade out of 10, to compare students at a glance. We have defined a number of such indicators, trying to capture most aspects of a student's online activity. The features we have selected are: the login frequency, the date of last login, the time spent online, the number of lessons read, the number of lessons downloaded as a PDF to read later, the number of resources attached to a lesson consulted, the number of quizzes, crosswords, assignments, etc. done, the average grade obtained in graded activities, the average last grade obtained, the average best grade obtained, the number of forum topics read, the number of forum topics created, and the number of answers to existing forum topics. For every "number of x" feature, we actually used a formula that would reflect both the distinct and total number of times that this action had been done.

From these indicators, we built by a weighted mean higher level ones representing a facet of learning, like online presence, study, graded activity, social participation and results. Then, at an even higher level but by the same process, a single general grade, which we called the General Learning Index and which gave its name to the application.

2.3 Machine learning

For a more complex output, we use different machine learning methods to analyse the data more in depth and interpret it semantically [1]. We use classical clustering and classification algorithms, in their implementation by the free library Weka.

We provide the following algorithms: for clustering, Expectation Maximisation, Hierarchical Clustering, Simple K-Means, and X-Means; for classification, Logistic Regression, LinearRegression, Naive Bayes and Multilayer Perceptron. They can be used with or without cross-validation, and the random seed and number of folds can be manually selected. For clustering algorithms where the number of clusters is not decided by the algorithm, we allow to select a fixed number of clusters.

We use our indicators listed in §2.2 as features for learning, and for the classification algorithms, we use the mean grade obtained at the final exams as the class feature. As an output, we obtain groups of students. In the case of clustering, we have to look at their indicators to understand the meaning of these groups. With classification, we can try to see which indicators can predict the final grade.

3. FUTURE WORK

We have already thought of new features that we would like to implement.

We want to compare the results obtained by all machine learning algorithms to see if one seems better suited. Later, we will also implement another HTM-based machine learning algorithm, and again compare results. We also want to add regression to try and predict the final grade.

Another facet that the data we have gathered could reveal is the quality of the study material: is a quiz too hard, so that students systematically fail it? Is a lesson less read than the others - maybe it is boring? Do the students feel the need to ask many questions in the forums?

We could partly automate the training managers' work by creating an intelligence virtual tutor that will directly interact with students and teachers. It could suggest students a next action based on their last activity and graded results, or also give them a more global view of where they stand by using the machine learning results. It could also send them e-mails to advise them to login more frequently or warn them that a new activity has opened. It could also warn the training manager of any important or unusual event.

4. CONCLUSION

This application uses data mining and machine learning methods to solve the problem of student monitoring in e-learning. We have detailed how the implementation allows to meet our goals by a good mix of different levels of granularity in the viewing of the data (raw data, statistics and data processed by different clustering and classification machine learning algorithms). Such a tool is very appreciated by our first users and is very innovative.

To do this, we have had to define indicators that serve both for statistics and for machine learning features. These indicators are both relevant to our project and generic enough to be of use for the community.

We will also present a poster in this conference to describe some preliminary clustering results obtained using this application.

5. REFERENCES

[1] C. Romero, S. Ventura, and E. García. Data mining in course management systems: Moodle case study and tutorial. *Computers and Education*, pages 368–384, 2008.