

# A method for generating features representing the students' degree of anticipation/delay to complete assignments

Francisco Cervera-Mercadillo,  
Juan A. Lara  
Madrid Open University, UDIMA  
panchitopin@gmail.com,  
juanalfonso.lara@udima.es

## ABSTRACT

Some current methodologies stress the importance of continuously assessing the students to check their progress, instead of having an only final examination. To do so, several learning resources are presented to the students, so they can complete different actions over those resources at certain particular dates. This scenario presents a good chance for research since it might be useful to analyze the point up to which students complete in time the tasks they are supposed to do. In this paper, we present a method that takes raw log Moodle data and generates new features that represent the degree of anticipation/delay of students when completing the tasks suggested by their tutor. We have developed a system that implements this method obtaining some minable sights that preliminarily seem to be useful to predict phenomena such as academic dropout. Obviously, future deep experiments must be conducted to demonstrate the validity of those new features.

## Keywords

Data preprocessing; E-learning; Continuous Assessment; Assignment-related temporal features; Moodle.

## 1. INTRODUCTION

In the last years, new formative paradigms such as E-learning (Electronic Learning) have emerged in order to provide people with ubiquitous learning [1]. E-learning platforms like Moodle provide useful data about students' behaviors that can be exploited by Educational Data Mining (EDM) or Learning Analytics (LA) techniques [2].

Many current education initiatives are based on continuous assessment during the courses. It means that students are encouraged to complete different assignments at certain suggested dates [3]. Some of those students will complete those assignments at the suggested date, others will do it earlier and some of them may complete it late.

In this paper, we present a method that takes Moodle logs from a particular course and a list of suggested dates where assignments are suggested to be completed by the students according to the

tutor recommendation and generates a minable sight in the form of a table containing as many rows as students enrolled in the course and as many columns as assignments or tasks that students should carry out. Each cell will take an integer value representing the degree of anticipation or delay for the particular student (row) to complete each task (column) suggested by the tutor. These new features can be useful in the prediction of the students' performance, which is one of the main goals of EDM.

Although there is some literature on procrastination [4], to the best of our knowledge, it is the first time that the high-level features we propose have been used in EDM field so our ideas may represent an interesting line of research, which is the main contribution of this paper.

The rest of this paper is organized as follows. Section 2 presents a brief description of the proposed method. Section 3 contains a technical description of the implemented tool and the preliminary results obtained. Finally, Section 4 includes a discussion of the results and conclusions obtained, as well as some potential future lines of research.

## 2. METHOD

In this paper, we propose a method that intends to generate new features from Moodle logs that represent the degree of anticipation or delay for each student to complete the tasks proposed by the tutor in order to reach the formative results supposed to acquire during continuous evaluation.

Those suggestions may be really diverse and include tasks such as reading a document, watching a video, taking part in a forum or submit a report. In particular, we have defined a series of potential kinds of tasks that the students can carry out with educational resources uploaded to the Moodle virtual classroom by the tutor. Those are: View, Create, Update, Delete, Subscribe, Review, Submit, and Start. For each educational resource and type of task, the tutor has to define a reference date when the students are supposed to complete that task on the respective resource (Table 1).

The proposed method intends to take raw Moodle data and generate a new feature for each resource and type of task in a way that each new attribute will take negative values for "eager" students, positive values for "late" students, and 0 values for students who complete their tasks exactly the same day as the tutor suggested.

This method is intended for EDM/LA experts who intend to use these kinds of features in order to analyze their impact on students' performance. Note that our method starts from Moodle

Francisco Cervera and Juan Lara "A method for generating features representing the students' degree of anticipation/delay to complete assignments" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 574 - 577

log data that register all the events (actions) carried out by the students on each resource of the Moodle platform.

**Table 1. Excerpt of an example of suggested tasks and dates**

Computer Architecture Course	
Week 1	
Task	Suggested Date
View the Presentation Session (VPS)	8-Oct-2019
Read the Teaching Guide (RTG)	13-Oct-2019
Week 2	
Task	Suggested Date
Subscribe to Doubts Forum (SDF)	14-Oct-2019
Visualize Unit 1 PDF Document (VU1)	16-Oct-2019
Submit the report of Activity 1 (SA1)	20-Oct-2019
...	...

We have also defined our approach in a way that the expert can perform the following tasks to particularly design his/her data analysis:

1. Student selection (focus only on some certain students).
2. Period selection (focus only on certain time intervals of the course).
3. Resource selection (only consider some educational resources).
4. Task type selection (only consider some types of actions for each resource).
5. Reference dates definition (define, for each selected resource and type of action).
6. Generate new attributes' values (for each selected resource, type of action and student).
7. Manually add the class attribute for the latter analysis (dropout, pass/fail, ...).

That would lead to a minable sight for predictive purposes with the newly generated features. Table 2 shows an example.

**Table 2. Excerpt of an example of generated minable sight for academic dropout prediction**

	New Generated Features						Dropout
	Week 1		Week 2			...	
Student	VPS	RTG	SDF	VU1	SA1	...	
<i>Student<sub>1</sub></i>	0	-2	0	-1	-3	...	0
<i>Student<sub>2</sub></i>	+2	+3	0	+8	+12	...	1
...	...	...	...	...	...	...	...
<i>Student<sub>n</sub></i>	+1	0	+2	-1	0	...	0

### 3. IMPLEMENTED SYSTEM AND RESULTS

In this section we will explain the main aspects about the design of the system (3.1), how the system works (3.2) and system outputs (3.3).

#### 3.1 Design and Technical details

The system was designed in order to meet four important main objectives:

1. The implemented method should guide the user step by step, executing the different tasks developed as an assistant, to finally obtain the desired minable view.
2. Due to the problem of dealing with large volumes of records to be processed, data persistence was decided not to be necessary. In our case, preprocessing the data in memory helped speed up the application of the different data selection, cleaning and transformation techniques.
3. During the development of the application, it was intended to design a system focused on usability, in order to facilitate the learning and use of the tool by the end-user.
4. It was necessary to define a properties file so the user could define some parameters difficult to provide a value for in execution time.

To achieve these objectives, the implementation was based on a web development paradigm with generally visual components (selects, multi-selects, calendars for date selection), except those values that are necessary to enter into the system for the creation of new high-level features, which inevitably force the introduction of the necessary values.

From a technical point of view, the implementation of the tool was carried out under the Java development language (JDK 1.8), mainly due to the multiple capabilities and features it offers. Since our system is a web application, we relied on its main framework, Spring 5.0, granting an agile development, based on the injection of dependencies and therefore, decoupled and easily scalable. This framework offers by default the Thymeleaf 3.0.4 template engine for the creation of the different views and, in our opinion, perfectly meets the needs imposed, as it allows the definition of reusable fragments and layouts, as well as a wide set of expressions to directly deal with the different data models generated at each step or task performed.

We also used the Bootstrap 4.0.0 framework, which has facilitated the design of the application interfaces, providing the web application with responsive features necessary to adapt the tool to the different existing viewing platforms. Apache Maven 4.0.0 let us manage and build the project in a simple way, as well as define the necessary dependencies of the system, obtaining them directly from its central repository.

Other technologies or frameworks such as jQuery (improvement of the interactivity of the application with the end-user), JUnit 5, Mockito (unit tests of validation of the most important methods of the application and integration of components) and Docker (creation of lightweight containers and highly portable for deployment) were used for the development of the project.

### 3.2 System working

As already mentioned, the system defines a series of tasks to be performed as an assistant. The system will guide the expert through the following implemented tasks, which are listed and detailed below, in order to be able to design a personalized educational data analysis:

1. Loading of the Moodle log file (Comma Separated Values -CSV- file) of a specific subject, that is, the source dataset to be preprocessed.
2. The expert selects those students not to be included in the final dataset.
3. The system allows the expert to select a date range, discarding all those records that are not inside that period. The system will not allow selecting dates outside the dates range existing in the dataset.
4. The expert selects those resources that he or she does not want to include in his analysis.
5. The system allows the grouping of resources, in order to consider them, from that moment, as the same resource.
6. The system shows for each resource, the different types of actions existing in the dataset, and the expert selects those that he or she wishes to remain in the final dataset (Figure 1 shows a screenshot of this step).

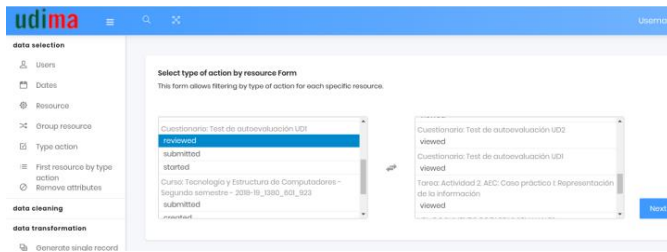


Figure 1. Selection of types of actions

7. The expert establishes the reference date for each resource and type of action (see Figure 2).

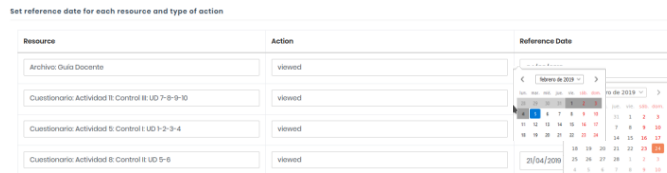


Figure 2. Reference dates definition

8. The system selects the first existing action type for each resource and user.
9. The system generates for each student a single record where, for each resource and type of action, it creates an entire attribute. This attribute represents the difference in days, between the reference date and the date recorded in the input file (see Figure 3, where students' names have been anonymized).

User	Sesion1	Read U1	Activity1
Student 1	1	65	47
Student 2	-2	7	14
Student 3	-6	-2	-4
...	3	63	42

Figure 3. Attribute values system generation

10. The system allows the expert to give value to some class attributes for classification purposes (see Figure 4).

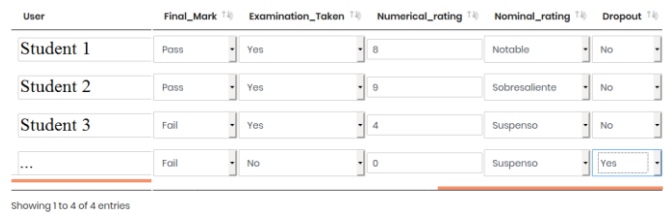


Figure 4. Class attributes user value definition

11. The expert exports the results obtained to a file with CSV format.

### 3.3 System output

The system generates a CSV file containing one record per user (anonymized), indicating the degree of advancement or delay in the performance of the activities selected by the expert, according to the date proposed by the tutor, as well as the new features created for classification. Figure 5 shows an example.

```
"User","Teaching Guide viewed","Activity11 viewed","Activity5 viewed","Activity8 viewed",-
"0","-3","1","65","47","3","-1","4","14","1","Pass","Yes","8.0","Remarkable","No"
"1","-2","-2","7","14","-1","11","-5","-2","-3","Pass","Yes","9.0","Outstanding","No"
"2","-3","-6","-2","-4","-2","2","2","-5","-3","Fail","Yes","4.0","Pass","No"
"3","-27","3","63","42","-1","2","21","8","90","Fail","No","0.0","Fail","Yes"
```

Figure 5. Example of CSV file generated by the system

After conducting some preliminary experiments, it was possible to obtain some interesting minable sights and get some predictive models using Weka<sup>1</sup>. Applying, for example, a classification algorithm based on decision trees, we obtain an interesting model to predict whether a student will pass or fail the course. Below, we show the results obtained for this example case:

```
Decision Stump-Classification
Case Study I_viewed <= -2.0 : Pass
Case Study I_viewed > -2.0 : Fail
```

If we look at the predictive model returned by Weka, we find that the activity “Case Study I” becomes especially relevant among all the resources and types of actions of the original dataset. It indicates that all those students who visualize this activity with an advance of two or more days, with respect to the reference date established by the tutor, will finally pass the subject. Otherwise, they will fail.

<sup>1</sup> <https://www.cs.waikato.ac.nz/ml/weka/>

This acquired knowledge could directly influence the decisions that teachers should make regarding the activity, monitoring their development through tutorials with students, reinforcing the teaching material, encouraging students to carry it out or other types of actions.

#### 4. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a method that allows generating new features that represent the degree of anticipation or delay of students when completing the tasks suggested by the tutor. Our method is applicable in open courses that adopt methodologies based on the existence of a tutor that guides students by defining a series of continuous tasks that students are encouraged to complete at a certain date.

We have developed a preliminary system that implements that method with Moodle data, and also gives the user the possibility of performing some preprocessing tasks, such as student selection, resource selection, and so on. We have conducted preliminary experiments to obtain some minable sights of different high education open courses. Those tests make us be optimistic about the usefulness of the newly generated attributes and their potential application for future research.

The main future line of research that we should carry out next is the application of the method in different courses and analyze if the new proposed features are valid to predict important educational phenomena such as students' dropout, students' final marks, and so on.

Once a stable release of the system is finished and tested, we intend to provide the community with an access URL so that the system can be publicly used and tested.

#### 5. REFERENCES

- [1] Cárdenas-Robledo, L. A. and Peña-Ayala, A. 2018. Ubiquitous learning: A systematic review. *J. Telemat. Inform.* 35, 5, 1097-1132. DOI=<https://doi.org/10.1016/j.tele.2018.01.009>.
- [2] Romero, C. and Ventura, S. 2020. Educational data mining and learning analytics: An updated survey. *WIREs Data Mining Knowl Discov.* e1355. DOI=<https://doi.org/10.1002/widm.1355>
- [3] Burgos, C., Campanario, M. L., de la Peña, D., Lara, J. A., Lizcano, D. and Martínez, M. A. 2018. Data mining for modeling students' performance: A tutoring action plan to prevent academic dropout. *Comput. Electr. Eng.* 66, 541-556. DOI=<https://doi.org/10.1016/j.compeleceng.2017.03.005>.
- [4] Hooshyar, D., Pedaste, M. and Yang, Y. 2020. Mining Educational Data to Predict Students' Performance through Procrastination Behavior. *Entropy* 22, 12. DOI=<https://doi.org/10.3390/e22010012>