Analyzing Student Behavior in Moodle System

Yassine Chajri
Sultan Moulay Slimane University
Beni Mellal
Morocco

Mohammed Chajri
Sultan Moulay Slimane University
Beni Mellal
Morocco

ABSTRACT
Educational data mining is an interesting discipline that focuses on developing methods to extract knowledge and discover patterns from online learning systems. This work is an application of data mining in learning management systems. Our objective is to introduce educational data mining by describing a step-by-step process using a variety of techniques such as Attribute Weighting, Classification, Clustering, and Association Rules to achieve the goal of discovering useful knowledge from Moodle. For association rules, we will present a comparison between two data mining algorithms, Apriori and FP-Growth, to justify our choice of the FP-Growth algorithm. Analyzing mining results enables teachers to better allocate resources and understand student behavior.

General Terms
Educational data mining.

Keywords.
Educational Data mining, E-learning, Data Mining, Moodle, learning patterns

1. INTRODUCTION
Data mining, or knowledge discovery, is a computer-assisted process that involves searching through and analyzing vast datasets to extract meaningful insights. Data mining tools predict future behaviors and trends, enabling proactive decision-making based on knowledge [1]. These tools scan databases for hidden patterns, uncovering predictive information that experts may overlook because it falls outside their expectations.

Thus, data mining refers to the extraction or "mining" of knowledge from large volumes of data using advanced techniques such as classification, clustering, and statistical analysis.

Educational data mining is an emerging field focused on developing methods to explore the unique types of data generated in educational environments and applying these methods to gain a better understanding of students and the contexts in which they learn. By leveraging data mining techniques, various types of knowledge can be uncovered, such as association rules, classifications, and clustering. The discovered insights can be used to predict student enrollment in specific courses, analyze the shift from traditional classroom teaching models, detect unfair practices in online exams, identify anomalies in student grade reports, predict student performance, and more.

The primary objective of this paper is to apply data mining methodologies to study student performance. Data mining offers numerous techniques that can be utilized to analyze student outcomes.

This paper is structured as follows: Section II defines key terms central to our work, Section III describes the data preparation and preprocessing steps, and Section IV presents the experimental results of the data mining algorithms.

2. EDUCATIONAL DATA MINING (EDM)

2.1 E-Learning
E-learning is a form of learning that typically involves using a computer to deliver part or all of a course, whether in a school setting, mandatory business training, or a fully remote course [2]. Initially, online learning received negative feedback, as many believed that introducing computers into the classroom would eliminate the human interaction that some learners require. However, as technology evolved, we have embraced smartphones and tablets in classrooms and workplaces, along with various interactive designs that make online learning not only engaging for users but also an effective method for delivering courses.

E-learning systems facilitate communication between students and teachers, resource sharing, content creation, assignment preparation, online assessments, and synchronous learning through forums, chats, news services, and more.

There are many online learning systems, such as Moodle, TopClass, Ilias, and Claroline. In our case, we will be using Moodle.

2.2 Moodle
Moodle "Modular Object-Oriented Dynamic Learning Environment" is an open-source learning management system (LMS) originally developed by Martin Dougiamas [3]. It is used by thousands of educational institutions worldwide to provide a structured interface for online learning or internet-based education. Moodle allows teachers to create online courses, which students can access as a virtual classroom. A typical Moodle homepage includes a list of participants (including the teacher and students) and a calendar displaying the course schedule and a list of assignments.

Other key features of Moodle include:

- Online quizzes.
- Discussion forums, where students can post comments and ask questions.
- Glossaries of terms.
- Links to additional web resources.

This makes Moodle a versatile and interactive platform for e-learning.

2.3 Data Mining Definition and Techniques
Data mining, also known as knowledge discovery in databases (KDD), refers to the extraction or "mining" of knowledge from large volumes of data. Data mining techniques are used to analyze vast datasets to uncover hidden patterns and

relationships that are valuable for decision-making [4]. Although data mining and knowledge discovery in databases (KDD) are often used interchangeably, data mining is actually a subprocess within the broader knowledge discovery process. The sequence of steps involved in knowledge extraction from data is illustrated in Figure 1.

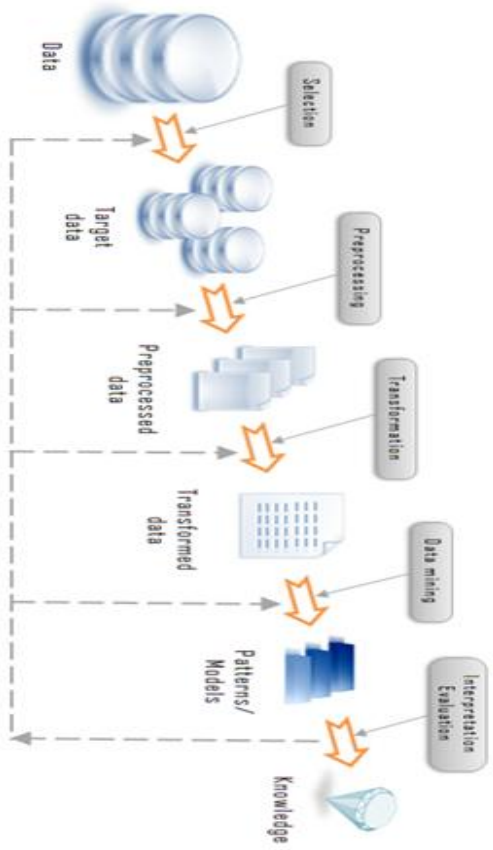

Fig 1: The steps of extracting knowledge from data

Various algorithms and techniques such as classification, clustering, regression, artificial intelligence, neural networks, association rules, decision trees, genetic algorithms, and the nearest neighbor method are used for knowledge discovery from databases. These data mining techniques and methods should be briefly introduced for a better understanding.

2.4 Educational Data Mining (EDM)

Educational Data Mining (EDM) was defined by Baker [5] as the scientific research field focused on developing methods to make discoveries from the unique types of data generated in educational environments and using these methods to better understand students and the environments in which they learn.

In other words, EDM is concerned with the development of methods for discovering knowledge from data originating from an educational setting. These data can be collected from historical and operational records stored in the databases of educational institutions or learning management systems (LMS). The online learning data mining process follows the same four steps as the general data mining process:

Data Collection: Student activities are stored in the LMS database; in our case, this refers to the Moodle database.

Data Preprocessing: The data is cleaned and transformed into an appropriate format for further analysis.

Data Mining Application: Data mining algorithms are applied to create and execute a model that discovers patterns and knowledge. Various data mining tools exist to apply DM algorithms; in this study, we will use RapidMiner.

Interpretation, Evaluation, and Deployment of Results: The obtained results are analyzed and used by instructors for further actions. The instructor can leverage the discovered insights to make informed decisions regarding students.

3. PRE-PROCESSING MOODLE DATA

Moodle is an open-source course management system that helps teachers create effective online learning communities. It is an alternative to proprietary commercial e-learning solutions and is distributed for free under an open-source license. The Moodle database contains several interdependent tables. However, we do not need all of this information, and it is also necessary to convert it into the required format used by data mining algorithms.

In the table (Table 1), we will therefore present the most important tables:

Table 1. Important Moodle tables

Table	Description
Mdl_assign	Assignment information
Mdl_assign_submission	Work done by student
Mdl_course	Course Information
Mdl_quiz	Quiz information
Mdl_quiz_attempts	Quiz attempts information
Mdl_quiz_grade	Quiz grade details
Mdl_forum	Forum information
Mdl_forum_post	Posts in forums
Mdl_forum_discussion	Discussions in forums
Mdl_forum_read	Posts reads by student
Mdl_grade_grades	Student grades
Mdl_log	Logs every user's action

3.1 Data Selection

It is necessary to choose the courses that may be useful to us. Therefore, we will select only the courses that use a higher number of Moodle activities and resources, including at least assignments, forums, and quizzes. In our database, we have selected 30 students and 4 courses.

3.2 Summarization Table

Our data is distributed across multiple tables, and a summary table has been created (see Table 2) that includes the most important information for our objective. This table (mdl_summary) provides a row-by-row summary of all the activities completed by each student during the course and the final grade obtained by the student in the course.

Table 2. mdl_Summary table

Attribute name	Description
Course	Identification of the course
Assignment_number	Number of assignments done
Quiz_number	Number of quizzes done
quiz_passed_number	Number of quizzes passed
quiz_failed_number	Number of quizzes failed
forum_posts_number	Number of posts in forum
forum_read_number	Number of reads in forum

total_time_assignment	Time spent on assignments
Total_time_quiz	Time spent on quizzes
Total_time_forum	Total time spent on forums student
Resource_view	Total Number of course materials and resources views
Final_mark	Student final grade

To create the mdl_summary table, we create a stored procedure with multiple queries.

3.3 Data Discretization

It may be necessary to perform discretization of numerical values to improve interpretation and understanding. Discretization divides numerical data into categorical classes that are easier for the instructor to understand (categorical values are more user-friendly for the instructor). All numerical values in the mdl_summary summary table have been discretized, except for the course identification number. For the discretized final grade attribute, three intervals and labels were used: (FAIL if the value is < 5, PASS if the value is ≥ 5 and < 8, and EXCELLENT if the value is ≥ 8).

For the other attribute, we used the equal-width method, which divides the attribute range into a fixed number of equal-length intervals. This method was applied to all other attributes using four intervals and labels: (ZERO, LOW, MEDIUM), and HIGH).

Table 3. Summary table (Categorical version)

Attribute	categories
Assignment_number	Zero,Low,Medium,High
quiz_passed_number	Zero,Low,Medium,High
quiz_failed_number	Zero,Low,Medium,High
forum_posts_number	Zero,Low,Medium,High
total_time_assignment	Zero,Low,Medium,High
Total_time_quiz	Zero,Low,Medium,High
Resource_view	Zero,Low,Medium,High
Final_mark	FAIL,PASS,EXCELENT

3.4 Data Transformation

The data must be transformed into the format required by the data mining algorithm. In our case, we used RapidMiner, which accepts various input types such as database tables, CSV, Excel, and ARFF files. For our study, we used database table inputs.

4. APPLYING DATA MINING ALGORITHMS AND INTERPRETING RESULTS

4.1 Information Gain

Before applying data mining algorithms, it is useful to determine the weight of each attribute. The Information Gain (IG) method is used for attribute weighting. IG provides a good indication of the degree of student involvement in a particular activity and is also used in the classification algorithm.

Table 4. Weighting with Information Gain IG

Attribute	Information gain
Total_time_quiz	0
forum_read_number	0
quiz_failed_number	0.010
forum_posts_number	0.015
total_time_assignment	0.370
quiz_passed_number	0.426
Quiz_number	0.484
Assignment_number	0.657
Resource_view	1

In Table 4, the resource view attribute has the highest weight (IG = 1.00), indicating that students frequently consult course materials. In second place, we found Assignment_number, while the third place is occupied by the number of quizzes completed by the student. At the bottom of the ranking, we see the total time spent on quizzes and the number of forum readings, both with IG = 0.

4.2 Classification

Classification involves predicting a certain outcome based on a given input [6]. To make a prediction, the algorithm processes a training set that contains a set of attributes and their corresponding outcomes, typically referred to as the target or prediction attribute. The algorithm attempts to identify relationships between attributes that can help predict the outcome.

RapidMiner provides several classification algorithms. In this study, we used the C4.5 algorithm to classify students as either passing or failing the course. C4.5 is an algorithm that generates decision trees and derives classification rules from the tree [7].

Our goal is to classify students into different groups based on their final grades, using the activities they completed in Moodle. We applied C4.5 to the mdl_summary table in its categorical version. As a result, we obtained a set of IF-THEN-ELSE rules derived from the decision tree, which provide valuable insights into student classification.

In our decision tree (see figure 2), if the Resource_view is null or low, the student is classified as FAIL. If the Resource_view is high and the quiz_passed_number is medium, the student is classified as EXCELLENT. If the Resource_view is average and the assigned_number is medium, the student is classified as PASS.

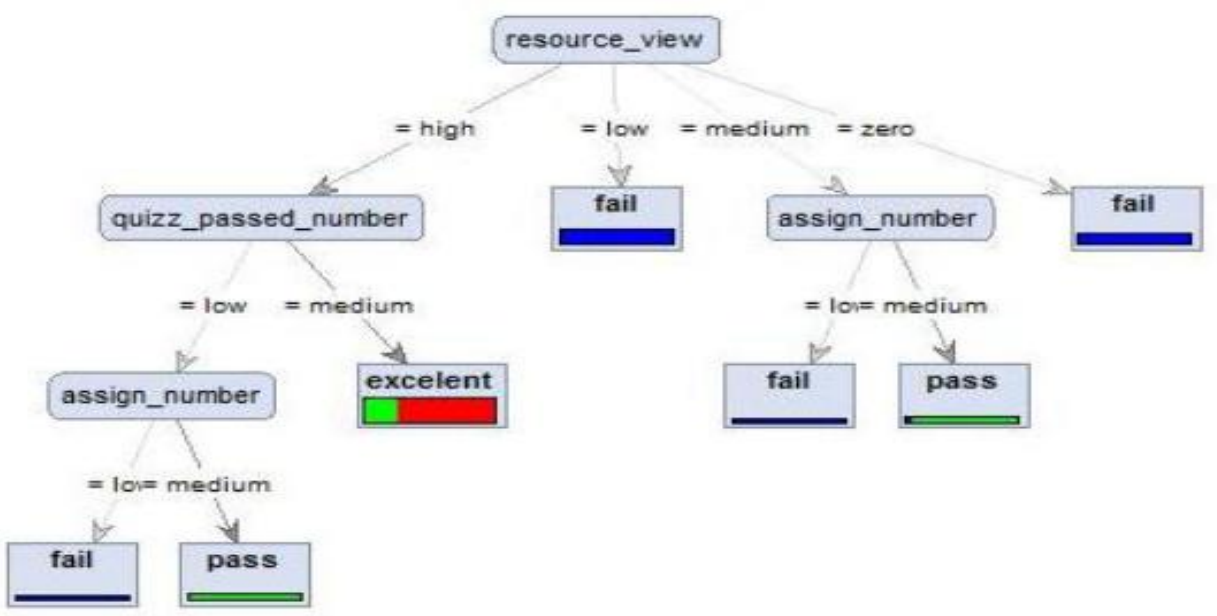

Fig. 2 Decision Tree

4.3 Model Validation

It is necessary to validate the model by applying validation and performance measurement techniques. By using X-validation, we obtained good results, such as an overall accuracy of 85.67%, and the other results are summarized in the table below.

Table 5. X-Validation results

label	Class recall	Class precision
fail	97.83%	100%
Pass	41.03%	91.43%
excellent	100%	73.10%

4.4 Clustering

Clustering can be defined as the identification of classes of similar objects. By using clustering techniques, we can further identify dense and sparse regions in the object space and discover the global distribution pattern and correlations between data attributes [8]. In e-learning, clustering has been used to: find groups of students with similar learning characteristics, promote group-based collaborative learning, and provide progressive diagnostics for learners [9]. RapidMiner offers several clustering algorithms. K-Means (MacQueen, 1967) [10], one of the simplest and most popular clustering algorithms, has been used here. It is an algorithm that groups objects into k-partitions based on their attributes. As a result, we have two clusters: an inactive group (cluster_0) and an active group (cluster_1).

Table 6. Clustering with k-mean algorithm

Attribute	Ctuster_0	Cluster_1
Assignment_number	2.067	4.244
total_time_assignment	185.644	15.011
Quiz_number	1.712	3.650
quiz_passed_number	1.951	3.783
quiz_failed_number	0.344	0.128
Total_time_quiz	506.436	330.689
forum_posts_number	0.282	0.406
forum_read_number	0	0
Resource_view	50.47	500

- Cluster_0 is characterized by inactive students in Moodle with a low number of actions in the system.
- Cluster_1 is characterized by active students in Moodle with a high number of assignments and resource views.

4.5 Association Rules

Association rule mining, one of the most important and well-studied techniques in data mining, was first introduced in [11]. It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories. Association rules are widely used in various fields such as telecommunications networks, market and risk management, inventory control, etc. Various techniques and algorithms for association rule mining will be briefly presented and compared later.

Association rule mining involves finding association rules that satisfy the minimum predefined support and confidence from a given database. The problem is generally decomposed into two sub-problems. One involves finding itemsets whose occurrences exceed a predefined threshold in the database; these itemsets are called frequent or large itemsets. The second problem is generating association rules from these large itemsets with minimum confidence constraints. Suppose one of the large itemsets is Lk, Lk = {I1, I2, … , Ik}, association rules with this itemsets are generated in the following way: the first rule is {I1, I2, … , Ik-1}⇒ {Ik}, by checking the confidence this rule can be determined as interesting or not.. Then, other rules are generated by removing the last elements from the antecedent and inserting them into the consequent. Furthermore, the confidence of the new rules is checked to determine their interest. These processes are repeated until the antecedent becomes empty. Since the second sub-problem is relatively simple, most research focuses on the first sub-problem.

The first sub-problem can be divided into two sub-problems: the process of generating large candidate itemsets and the process of generating frequent itemsets. We call itemsets whose support exceeds the support threshold large or frequent

itemsets. Itemsets that are expected or hoped to be large or frequent are called candidate itemsets [12].

4.5.1 Apriori

The Apriori algorithm is one of the most influential algorithms used for mining association rules, which was proposed by R. Aglawal et al. in 1994. According to the principles of the Apriori algorithm in [13], it is composed of two steps, one is extracting all the frequent itemsets; the other is generating all the strong association rules from frequent itemsets [14]. In fact, the essence is to iteratively generate the set of candidate itemsets of length (k+1) from frequent itemsets of length-k and check their corresponding occurrence frequencies in the database to obtain frequent itemsets of length (k+1) at each level. Therefore it can be seen that there are two main reasons to low efficiency of the Apriori algorithm: It is required to generate lots of candidate itemsets for generating each frequent itemsets; It is essential to scan database many times for generating each frequent itemsets [15].

4.5.2 FP-Growth

One of the currently fastest and most popular algorithms for frequent item set mining is the FP-growth algorithm [16]. It is based on a prefix tree representation of the given database of transactions (called an FP-tree), which can save considerable amounts of memory for storing the transactions. The basic idea of the FP-growth algorithm can be described as a recursive elimination scheme: in a preprocessing step delete all items from the transactions that are not frequent individually, i.e., do not appear in a user-specified minimum number of transactions. Then select all transactions that contain the least frequent item (least frequent among those that are frequent) and delete this item from them. Recurse to process the obtained reduced (also known as projected) database, remembering that the item sets found in the recursion share the deleted item as a prefix. On return, remove the processed item also from the database of all transactions and start over, i.e., process the second frequent item etc. In these processing steps the prefix tree, which is enhanced by links between the branches, is exploited to quickly find the transactions containing a given item and also to remove this item from the transactions after it has been processed [17].

4.5.3 Comparative study of Apriori and FP-growth algorithms

Apriori and FP-Growth are two well-known data mining algorithms used for association rule generation within a database. The main commonality between these two algorithms is their reliance on generating frequent itemsets to identify association rules. However, Apriori requires multiple database scans, generates a large number of itemsets, and recalculates the support for each itemset during each scan. Managing this vast number of itemsets is computationally expensive, as the frequency of each itemset needs to be tested repeatedly. In contrast, the FP-Growth algorithm aims to minimize the number of database scans, significantly reduce the number of itemset generations, and streamline the support calculation process. To achieve this, FP-Growth employs a divide-and-conquer strategy, breaking down the data mining tasks. This is why it uses the Growth Pattern Fragment approach, which avoids the costly candidate generation and testing process employed by Apriori.

Based on this comparison, we chose to use the FP-Growth algorithm, as its results—sets of frequent itemsets—serve as input for the Create Association Rule operator. This operator then generates several association rules, the most significant of which are presented in Table 7.

Table 7. Most important Association rules

Condition	Result
assign_number = low	final_grade = fail, quizz_passed_number = low
final_grade = fail	assign_number = low, quizz_passed_number = low
quizz_passed_number = low	assign_number = low, final_grade = fail
assign_number = low, final_grade = fail	quizz_passed_number = low
assign_number = low, quizz_passed_number = low	final_grade = fail
final_grade = fail, quizz_passed_number = low	assign_number = low
quizz_number = medium, resource_view = high	quizz_failed_number = zero, final_grade = excellent

5. CONCLUSION

In this research, a data mining model for Moodle data was proposed, incorporating several techniques: Attribute Weighting (Weighting by Information Gain), Clustering (K-Means), Classification (Decision Tree), and Association Mining (FP-Growth, Create Association Rule). Through educational data mining, educators can apply clustering techniques to identify distinct groups of students. These groups can then be used to build a classifier to categorize students. The classifier identifies the key characteristics of students in each group, enabling the classification of new online students. Lastly, instructors can utilize association rule mining to uncover potential relationships between these characteristics and other attributes. These rules not only assist in classifying students but also help detect the sources of any inconsistent values observed in student data.

Characteristics and other attributes. These rules not only assist in classifying students but also help detect the sources of any inconsistent values observed in student data.

6. REFERENCES

[1] Djamel Abdelkader ZIGHED, Ricco RAKOTOMALALA: Extraction des Connaissances à partir des Données.

[2] Alejandro Pena-Ayala, Educational data mining: A surveyand a data mining-based analysis of recent works.

[3] Martin Dougiamas, How we built a community around open-source software.

[4] Brijesh Kumar Baradwaj, Saurabh Pal ; Mining Educational Data to Analyze Students Performance in International Journal of Advanced Computer Science and Applications, Vol. 2, No. 6, 2011 .

[5] Baker, M.,(2010). Data Mining for Education. In McGaw, B., Peterson, P., Baker, E. (Eds.) International Encyclopedia of Education (3rd edition), vol. 7, pp. 112-118. Oxford, UK: Elsevie.

[6] Fabricio Voznika, Leonar Doviana, Data mining Classification.

[7] Nicolas Baskiotis , Michèle Sebag, C4.5 Competence Map: a Phase Transition-inspired Approach.

[8] Jiawei Han, Micheline Kamber and Anthoney K .H.Tung, Spatial Clustering Methods In Data Mining : A Survey in School Computing Science, Simon Fraser University.

[9] Cristóbal Romero , Sebastián Ventura, Enrique García : Data mining in course management systems: Moodle case study and tutorial.

[10] Vance Faber : Clustering and the Continuous k-Means Algorithm.

[11] Sotiris Kotsiantis, Dimitris Kanellopoulos,Association Rules Mining: A Recent Overview ; GESTS International Transactions on Computer Science and Engineering, Vol.32 (1), 2006, pp. 71-82.

[12] Agrawal, R., Imielinski, T., and Swami, A. N. 1993. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, 207-216.

[13] R. Agrawal, T. Imelinski, and A. Swami, "Mining Association Rules Between Sets of Items in Large Database," Proc. ACM-SIGM OD International Conference, pp. 208-216, 1993.

[14] A. Salleb and C. Vrain, "An application of association rules discovery to geographic information systems," Proc. The 4th European Conference on Principles of Data Mining and Knowledge Discovery PKDD, pp. 613-618, 2000.

[15] Y Jaya Babu, G J Phani Bala, Siva Rama Krishna T Extraction Spatial Association Rules From the Maximum Frequent Itemsets based on Boolean Matrix : International Journal of engineering Science & Advanced Technology Volume - 2, Issue - 1, 79 – 84.

[16] J. Han, H. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In: Proc. Conf. on the Management of Data (SIGMOD'00, Dallas, TX). ACM Press, New York, NY, USA 2000.

[17] Christian Borgelt, An Implementation of the Fpgrowth Algorithm in Department of Knowledge Processing and Language Engineering School of Computer Science, OttovonGuerickeUniversity of Magdeburg Universit atsplatz 2, 39106 Magdeburg, Germany.