*Master in Free Software (UOC)*

# Final MSc Project in Basic Research

## Using Free Data Mining Software and Clustering Algorithms to find Predictors from Student Qualifications

**Author**
*Oriol Boix Anfosso*
*(oboix@uoc.edu)*

**Tutors**
*Jose Antonio Morán Moreno (UOC)*
*Germán Cobo Rodríguez (UOC)*

22/06/2010

*Aquest projecte no hagués estat possible sense el suport de la meva família.*

*Tant de bo algun dia li pugui dedicar tant de temps i esforços a ella com n'he dedicat a aquest projecte.*

# Table of Contents

# Free Software tools for Data Analysis and Knowlegde Discovery in e-learning: State of the Art

## Introduction

As mentioned in a previous draft paper, this M.Sc. research project is aimed to elaborate a 'State of the Art' report concerning free software for data analysis. But, without loosing sight of this goal, I must restrict its scope. In the last weeks and while I've been searching for such tools and looking for information regarding their characteristics over the Internet, I realized that there really are a lot of free software tools related with all kinds of data analysis, from classical statistics  complete software applications (see R project, [1], for instance) to specific-goal software limited tools (for a concrete set of data mining methods or algorithms, for example, as could be said of classification, clustering and of the other KDD techniques -an exhaustive list of such software tools, including proprietary ones and large suites, complete applications, etc., for data mining, classical statistics, data warehouse, business intelligence solutions, etc. can be found at [2]). Thus, due to the large amount of such tools and the wide scope of the data analysis concept, I decided to make this project focused on data mining in e-learning, and some of the free software tools or applications which are relevant to this field. Indeed, an alignment of this project with the goals of the doctoral research line of data analysis at the UOC is accomplished. At this point I may recall that the improvement of the virtual learning campus (VLC) is the main goal of several research projects in this virtual university.

Once the project cut has been justified, from general data analysis to data mining in e-learning, from now on I proceed as follows: first of all, a brief summary of the state of the art of data mining in e-learning is a must to investigate and to expose. Then, once a general view of the state of the art (in the data mining in e-learning field) is known, we have a starting point from where to revise the related free software tools and applications. While this project goes ahead, we shall decide which of these tools will be used to put in practice some of the specific tasks in data mining. (I can not discard the possibility of making my own tool, or a support tool, if none of the currently known within the elaborated inventory does not fit well enough to our purposes). The  specific tasks in data mining will also be related with some derivative goals in the e-learning field, starting from data obtained from UOC students, their qualifications and their interactions with the VLC. Thus, putting the software tools in use under real study cases, we will be able to evaluate both of them (tools as well as the study cases).

## Data mining in e-learning

Data mining in e-learning is by itself a wide and emerging field in which a lot of research has been done and is continue doing nowadays. It is just an application of data mining in a specific domain, so from an upper point of view, data mining as a whole is also and definitely one of the fastest growing areas in computer science. It actually, as well as potentially, offers a lot of applications, powerful tools to analyse the many large databases used in every specific domain, as in business, in science and in industry.

Data mining is also known as *knowledge discovery on databases* (KDD) and is a branch of the most generic data analysis field, which its main goal is to extract knowledge from the vast amount of data (large data sets) contained in every kind of databases. As stated in [3], the general experimental procedure adapted to a typical data mining process involves the following phases: *state the problem and formulate the hypothesis*, *collect the data*, pre-process *the data*, *estimate the model* and, finally, *interpret the model and draw conclusions*.

We may pose an example of a problem to be modelled in the domain of e-learning: It may be interesting, for different reasons, to find homogeneous groups for students according to their achievements and qualifications. This way, effective learning groups could be established picking up an equally proportional amount of students from each homogeneous group, i.e., creating new heterogeneous groups of students. We formulate here an hypothesis related with the domain of study (the e-learning field), namely, that those mentioned heterogeneous groups will enhance the learning effectiveness in front of randomly created groups. Once the data mining process is finished and after an evaluation of the learning effectiveness of the newly heterogeneous groups, we may establish, at least at a certain degree, whether that hypotheses matches reality (this is simply an application of the scientific method, where an hypothesis can achieve the theory status through experimental results and testing). But we must not confound this specific-domain hypotheses with a data mining hypotheses: there is not such hypotheses to formulate in a data mining process, because we simply can not state anything before the data mining process is finished; the data mining process is just a knowledge discovery process, so the knowledge in any form is not still available in this first phase. Otherwise said, the human has nothing to say, but letting the data sets "speak for themselves".

Collecting the data is the next phase, and we can distinguish here two types of data collecting methods, concerning the user degree of implication in them: the *intrusive* way (active user participation) or the *non-intrusive* (passive user contribution -more on this later). Regardless of this classification of collecting data methods, which is appropriate for the e-learning domain (see an example of its relevance in [4]) there are other possible classifications if other factors are considered (information systems used, type of domain, observational versus experimental approach, etc.).

But data can be collected from a large variety of data sources, which includes every different type of databases (relational, object-oriented, multidimensional and online analytical processing, deductive, parallel, distributed, etc., as it is described in [5]), and also from other types of sources not specifically being databases but which can play the role of an implicit data source, as for instance (see [6]) web server log files (for web usage mining), text content and multimedia content (web content mining) and HTML, XML tags for extracting DOM structures (web structure mining).

This heterogeneity of data sources, plus other handicaps in the processes of collecting data, as for example features or attributes with blank or non-coherent values in some way, and also the necessity to adapt the available data sets to the best fit to our data mining problem, all of these are reasons why the data must be preprocessed always before applying them to the chosen data mining technique (in the next phase, to estimate the model). In generic data mining there are quite different tasks for preprocessing data (outlier detection and removal, feature selection, discretization of numerical attributes, scaling or normalization and encoding features, dimension reduction for large data sets, etc.) and almost everyone of them should be applied in the e-learning domain for either problem to be modelled or other; but in the e-learning domain there are some specific data preprocessing tasks specially appropriate to the characteristics of the sources for data collection in this domain.

E-learning is a global concept which entails the inclusion of every online learning, training and educational systems, which is referred as a new form of knowledge delivery through the advent and evolution of the World Wide Web**.** Thus, motivated by the web implementation of such e-learning systems and platforms, web mining is commonly used in the e-learning domain. Web mining is the application of data mining techniques to extract knowledge from web data, including web documents, hyper-links between documents, usage logs of web sites, etc. ([6]), and it has already been extensively applied to others domains such as e-commerce. In turn, web mining can be classified in (see [6], [7]) *web content mining* (the process of extracting knowledge from the contents of web documents - text mining and multimedia mining applies here), in *web structure mining* (the process of discovering structure information from the Web) and in *web usage mining* (to discover interesting usage patterns from web data -logs-).

Although it has been demonstrated in [4] that hybrid methods for constructing recommender systems achieve better results than those which are solely based upon one type of web mining (hybrid methods take into account knowledge obtained from web content, as well as from web structure and web usage), it is the *web usage mining* the one which has been more widely studied, specially due to the relative easy of treatment for the given data source, the web server logs (text content, for instance, can also relatively easily be treated but multimedia content can not). At this point we can retake the preprocessing data explanation and add an example of it regarding the web usage mining and the log files as the data source.

As it is explained in [8], *data filtering* and *data transformation* are two of the stages in pre-processing data from log files. In [9] we can read a clear example of such stages, applied to logs from the UOC VLC (virtual learning campus) usage. These logs are automatically generated by the Apache web servers and initially include a lot of no useful entries, as the load of icons, images, CSS, banners, etc. These lines must be removed, as well as any others with also no useful information, for instance those without user or session data, or finally those not related with the targeted students in the current study. An example of data transformation is also explained when the privacy issues are addressed: the logs have encrypted strings carrying user and session information, which are substituted with a new user and session identifiers that can not be traced back to the original users. (Although this data pre-processing is carried out, we should take into account that the log data files obtained at the end of the process are still huge in size, normally of quite some gigabytes). Another very good explanation for data cleansing in web usage mining can be found at the second paragraph (*2- Web Log Cleansing*) in [21].

Once the data is preprocessed, as stated above the phase of estimating the model is the next. The goal of this phase is to apply the chosen data mining techniques, algorithms or methods to the preprocessed data, to construct a model from which we may draw some conclusions about the stated problem or the formulated hypotheses in our domain or field of study (in the first phase). Thus, the model has to be interpretable to our purposes, i.e., to offer some useful knowledge about our domain goals. For example, if our domain problem is about to establish the relationship, if any, between the browsing behaviour of students and their qualifications, then we firstly need to construct a *descriptive* data mining model to find patterns in browsing behaviours, and after that to apply another descriptive data mining model to establish the dependency between those patterns found and the qualifications (in these cases, association rules apply here, but other descriptive models can be used, as for example unsupervised clustering [7]). But, on the other hand, if our domain problem consists in formulating an hypotheses, then we should construct a *predictive* data mining model to evaluate it (for instance, classification and regression trees apply here). An example could be that we suspect that a student who has not visited the campus room for delivering activities not submitting them, then he or she has a great probability to fail the final assessment. Thus, the predictive data mining model should predict that fact based on the training data from the current semester, and correctly classify  those students who has not visited that campus space belonging to the class of unsuccessful students (this is an obviously example, but it also applies to non-obvious ones, confirming or not the hypothesis -for instance, the prediction of students' scores upon non-trivial web usage features is carried out in [31]).

Apart from constructing a model for obtaining *new* knowledge or confirming an hypotheses related with our domain, another goal that usually arises within a data mining research is the comparison of the effectiveness and the efficient of the different data mining techniques or algorithms, including testing new ones. Indeed, the *Artificial Intelligence* area often intersects here, because the community is always looking for new algorithms, new approaches to increase their applicability and efficiency, and Artificial Intelligence provides a good theoretical base –as an example [32] offers a detailed theoretical study about how some kind of Bayesian inductive algorithms can be applied to infer and predict). Thus, the '*State of the Art*' in the field of *data mining in e-learning* concerns not only about research in new domain-related problems, and suggesting possible hypotheses to us, with always the main goal in mind of the improvement of the e-learning processes and the enhance of the quantity and quality of successful learners, but also with the improvement of the data mining techniques, algorithms and methods specifically applied to this field. Some illustrative examples of the '*State of the Art*' in the field of data mining in e-learning, additionally to all what is already explained, are shown below.

One of the domain problems some researchers have focused on is the personalization of a space for collaborative learning. These spaces are usually implemented over a distributed platform and the personalizations, which are based on the students profiles, are achieved by intelligent agents. In [10] fourteen intelligent applications for collaborative learning are described, from, for instance, ALFanet which offers a dynamically personalization of web pages (one of the oldest problems related with web mining and adaptive web learning systems), to others, like CASSIEL, which can also assist students in configuring the discussion groups and the learning planning.

Besides the collaborative learning over distributed platforms, there is another data mining concept that may be applied to a generic improvement of the e-learning systems: the distributed data mining techniques. These techniques are proven to be valid and useful [11] addressing the mentioned domain problem, namely, the problem of disposing of a valid model for a generic e-learning system or platform, i.e., given the variety of such systems, a predictive model that can be applied for inferring future behaviours independently of their differences. Thus, in [11] the CIECoF (*Continuous Improvement of E-Learning Framework*) is presented as a general framework for addressing the mentioned domain problem. This framework consists of applying some unsupervised methods, as association rules, to generate recommendations for the improvement of learning courses. But despite of some other interesting conclusions, as the use of a improved *Apriori* algorithm with a derivative called *Predictive Apriori* (it does not need prefixed values for the support and the confidence, it just looks for the N specified number of association rules maximizing the probability of making a right prediction from the data set) (see also [23]) -and the evaluation of this algorithm fits in the above mentioned set of usually research objectives, namely those related with the study of efficiency and effectiveness of algorithms- for reaching out its aim of being useful for a generic e-learning system, this framework makes use of an intrusive way of collecting data: a voting system (similar to those used in e-commerce web applications). At this point, I consider very interesting to compare this study [11] with that mentioned above [4] because of the contrast of priorities (the later prioritizes the fact that the way of collecting data from e-learning systems should be non-intrusive, as the characteristics of participants in this domain are very different to those of the participants in other areas such as e-commerce).

But disposing of a valid model for a generic e-learning system, as it has a higher and implicit degree of difficulty which can be partially compensated with the active participation of learners in providing data, has not aroused as much interest in researchers as addressing domain problems for e-learning systems with a specific set of characteristics or functionalities, where learners mainly assume a passive role. There are nowadays a lot of entities and organizations, as well public as private, offering education and formation services through the web. Several of such education services make use of some kind of LMS (*Learning Management System*), software applications that are found as well as privative (WebCT, Virtual-U, etc.) as free and open source (Moodle -see an example in [12]-, Atutor, etc.). Other institutions use some kind of tailored e-learning platform adapted to its own designed learning planning and attending other issues as those related with marketing interests. A very known example of that is the UOC virtual campus [13], which is also designed following some recognized educational standards for e-learning, as SCORM [14].

The main problem of LMSs is that they are static, meaning that the learner can choose a browsing path through the course and that path could not be the most effective according the interests, the knowledge and the needs of the particular learner. A first solution to this problem was provided with the introduction of the adaptive (educational) hypermedia web systems (for instance, Interbook, ELM-ART, AHA! [15]). From static LMSs we can mainly address domain problems related with some kind of domain predictions or domain hypotheses as, for instance (see [16]) it is the problem of predicting a student's final performance based on his or her work (submitting activities, forum participation, etc.) in a VLE (virtual learning environment). Instead of this, working with adaptive systems or with some tailored ones with the capacity of dinamically generating web objects or the like, allow the researchers focus their goals in constructing some kind or recommender systems, or systems that dinamically adapt to students, or that which also dinamically provide some feed-back to teachers that let them adapt the courses they design to achieve a better learning response from learners.

Some of the particularities belonging to a data mining process applied to the Moodle LMS can be found in [16] and, mainly, in [17]. In [16], as mentioned above, the approach is to take a domain problem as a starting point (predicting a student's performance based on his or her work), and after an explanation of why the chosen data mining technique, a MIL (*multiple instance learning* [18]) derivative, fits in modeling the stated problem, it is applied to a case study through the Moodle LMS. Using Moodle, the students can consult and submit assignments, try to pass designed quizzes and also read o write in forums. Based on these different activities, a MIL technique is used as a wrapper for modeling the domain problem: the students are treated as patterns which are bags in the MIL representation, with three identifying attributes (user id, course id and the final mark obtained -or predicted to obtain- by the learner in such course), and the different activities are modeled as instances inside every bag, with the corresponding values for each student depending on the activities done. Note that different courses may have different number and types of activities, and different students may also do a different number of such activities. Thus, every bag is flexible enough to deal with such variability in activities through its instances. From a training data set where, as explained, data is modeled through a MIL representation, we should choose an algorithm to generate the classification rules and create at least a prediction rule to determine whether a student will pass the course successfully or not, depending upon the activities done. As explained above, one of the main goals in research in data mining is always to find better algorithms to reduce the time and space needs for computing problems, including the use of approximative solutions and heuristic approaches to deal with high complexity or hard problems. In this way, [16] is also a good example because it proposes a new genetic evolutionary algorithm, called G3P-MI, to find out the best classification rule from a initial population of such rules (which is optimized through recombination and mutation processes) and then compares it with other types of MIL-compatible algorithms such as PART and Bagging (supervised learning), MILR (logistic regression) and others (table 4 in [16] shows the results obtained for these algorithms in the case study).

In [17], the collecting and preprocessing data particularities for the Moodle LMS are explained, as for instance in this platform data sets do not come from log files but from a relational database. Then, from this preprocessed data, the application of different data mining techniques (classical statistics, visualization, clustering, classification, association rules and others) and some possible domain problems which can be addressed with such techniques are also explained.

Aside from research related with LMSs and specific domain problems or domain hypotheses, there is a lot of research about personalizing e-learning systems based on the students profile. This personalization includes the recommender agents (see [4], [11], [19] and [20], and Table 1, page 62 in [8] for a generic reference to the main filtering methods in constructing recommender systems) and other features of a web site that may be personalized for improving the provided utility as a learning service, namely the web site contents, the individual design of its pages and also its general structure ([22]). Personalizing these three mentioned features often belongs more to a teacher-related approach than to a student-related one. That is, to extract knowledge using web mining techniques to provide some feedback to teachers and / or authors that allow them adapt the courses they design to achieve a better learning response from learners. In [23] a methodology for addressing this domain problem is proposed: the CIECoM (*Continuous Improvement of E-learning Courses Methodology*): in a first stage the *adaptive* course is constructed by the teacher (the author) providing its contents and its structure; then the course is published within a web server and, in the next stage, the course is executed, i.e., it is used by students; while the course is running the server log files are generated in a non-intrusive way and after collecting and preprocessing the log data, the CIECoM applies a or some data mining algorithm(s) to detect possible problems in the course initially designed, so there is a base where to construct recommendations from to the authors for modifying the course contents or the course structure in a proper way. The CIECoM methodology includes a module for knowledge discovery in the form of association rules and, as mentioned before, it uses the modified Apriori algorithm called Predictive Apriori. For more detailed theoretical details I advice you to read such reference [23], as well as for those interested in the study case, which is based on an adaptive learning tool called INDESAHC ([24] and [34]), which permits the construction of adaptive hypermedia courses compatible with the Moodle LMS. Related with the same domain problem and with the same e-learning systems (the adaptive hypermedia -with AHA! in this case) [25] reports a more wide research study in comparing different algorithms and data mining techniques for providing feedback to the teachers and authors. In brief, it shows us a study case where a comparison between some of the most usual algorithms (ID3 -decision trees-, Apriori -association rules-, Prism -inductive algorithm-) and the proposed GBGP (an evolutive or genetic algorithm) is carried out.

Plus the research related with LMSs and adaptive systems, as mentioned before there is also research regarding tailored e-learning platforms as the UOC VLC (virtual learning campus). In particular, the UOC campus follows the LOM and SCORM standards and some research in web usage mining is already carried out [9], and more is currently in progress. As an example, in [9] the researchers try to find out whether or not in the UOC virtual environment some type of relationship between the learners' navigational behavior and their qualifications can be established (taken into account other students' features as age, gender, total of course credits, etc.). The TwoStep algorithm (unsupervised clustering) is used here and, once more, the main goal is to extract knowledge to help the authors and teachers improve the learning process in some way. Another study case worked out with the help of using the virtual learning campus of a university (the University of Taiwan in this case) as the experimental e-learning platform, can be read in [26]. This is just another example where web-based instruction and learning recommendations and personalization can be addressed with association rules or clustering and, in this case, the emfasis is put in a FP-growth (frequent pattern) derivative method (the cross-level FP method, see [26] for details) as an alternate to others as the Apriori's derivates mentioned above.

Data mining in e-learning is a young discipline that is continuously growing every day. In [27] and, specially, in [28] we find two very good summaries of the state-of-the-art in this discipline ranging its first ten years of life (from 1995 to 2005). But many advances have been made in the last years (some of them already mentioned in this report, and others referenced or cited by references), and a lot of research lines are currently in progress, trying to address different specific domain problems but always with the main goal in mind of providing improvement to the e-learning processes. Additionally, as *web mining* is the area in data mining which best fits in the major part of e-learning domain problems, every researcher in the e-learning domain should have a deep knowledge in this area and in its advances ([29], [30]), which is indeed more mature because it was initially applied to other older domains as e-commerce (there are a lot of advances in web mining, for instance and as a final example, in the sixth article, p. 115, from the later reference, [30], it is explained how a constraint that always arises with the problem of discovering classification rules from Relational Data, namely the acyclicity of the induced Bayesian networks by the models used, can be solved introducing Markov networks instead).

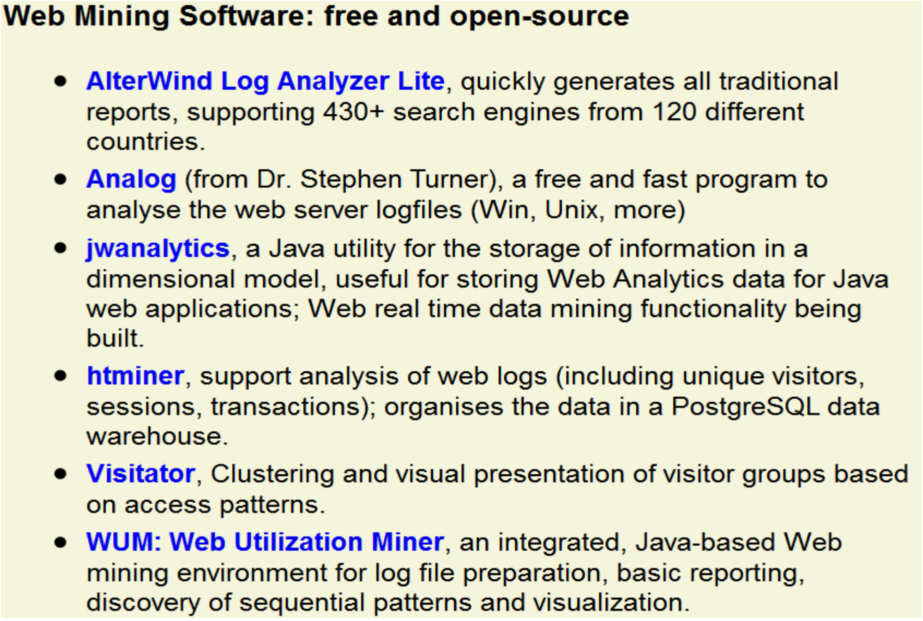
## Free Software for Knowledge Discovery

There is a lot of free software LMSs, as Moodle or ATutor, and also free or open source adaptive learning hypermedia tools, as SALMS ([39]) or as the

INDESAHC implementation ([24] and [34]) (which is free software–based, as every software developed by the CPMTI [35] and formerly by its predecessor, the EATCO -a research group belonging to the University of Córdoba [33]). But we are not interested in these kind of free software tools (e-learning tools), but in those which are targeted to perform data preprocessing to collected data, data mining techniques to extract knowledge or those that help us in visualizing and interpreting the results. For instance, the EPRules tool ([25] and [32]) is a good example of a free software tool specifically developed by the own researchers to address the corresponding particular study cases of their researches. EPRules is oriented to be used by non-experts in data mining, and to discover knowledge from SHAEW systems (*"Sistemas Hipermedia Adaptativos para la Educación basada en Web"*) as INDESAHC, in the form of prediction rules.

Regarding other specific and uncommon software tools (as EPRules is), possibly developed by the own researchers, through the "Red Española de Minería de Datos y Aprendizaje" [36] (*Spanish Data Mining and Learning Network*) it is possible to contact different research groups in the data mining domain, from different Spanish universities, and to find out if any of them is currently dealing with the development of a still unknown free software data mining tool.

Aside from these possibilities, there exist quite data mining tools in the free software world, from the most complete suite, Weka [37] (*Waikato Environment for Knowledge Analysis*) with an integrated graphical interface and allowing the most data mining techniques to be applied, to other smaller tools specifically developed to deal with some of the areas or method sets as every mentioned along this state-of-the-art report, namely and for example, for classification, or clustering, or for statistical analysis, text mining, web usage mining, data cleansing, etc. A good resource where to find a lot of software solutions, including privative and open source, is [2]. We must be careful when looking for free software solutions here, because in this web site the free software is not always clearly distinguished from the freeware software.



**Web Mining Software: free and open-source**

- **AlterWind Log Analyzer Lite**, quickly generates all traditional reports, supporting 430+ search engines from 120 different countries.
- **Analog** (from Dr. Stephen Turner), a free and fast program to analyse the web server logfiles (Win, Unix, more)
- **jwanalytics**, a Java utility for the storage of information in a dimensional model, useful for storing Web Analytics data for Java web applications; Web real time data mining functionality being built.
- **htminer**, support analysis of web logs (including unique visitors, sessions, transactions); organises the data in a PostgreSQL data warehouse.
- **Visitator**, Clustering and visual presentation of visitor groups based on access patterns.
- **WUM: Web Utilization Miner**, an integrated, Java-based Web mining environment for log file preparation, basic reporting, discovery of sequential patterns and visualization.

*Figure: Screen cap from [2] showing us some free software in web mining.*

Another free software tool that I want to emphasize is Tanagra [38]. This tool includes the majority of the data mining techniques, as statistical and automatic learning, supervised algorithms, clustering, parametric and non-parametric statistics, association rules, decision trees, and so on. It is specifically designed for academic purposes and thus, it does not include commercial features as connection with datawarehouses, graphical interactivity and the like. Instead, it is compatible with Weka.

As a final conclusion concerning free software, for this M.Sc. research project I firstly need to establish the domain problem(s) that is going to be addressed, and then determine and find out which of the available free software tools best fit in the related study case(s). (More on *Free Software for Knowlege Discovery* later).

# Using Free Data Mining Software and Clustering Algorithms to find Predictors from Student Qualifications

## 0- Abstract

In this paper a researh both in finding predictors via clustering techniques and in reviewing the DM free software is achieved. The research is based in a case of study, from where additionally to the DM free software used by the scientific community, a new free tool for preprocessing the data is presented. The predictors are intented for the e-learning domain as the data from where have to be inferred are student qualifications from different e-learning environments. Through our case of study not only clustering algorithms are tested but also additional goals are proposed.

## 1- Introduction

In the last decade the number of e-learning services offered over the Internet has grown exponentially, thus the interest for making the e-learning process better has also been increasingly attended by domain experts and scientific researchers. But this enhancement can not only be understood as an improvement of the IT resources (more capable servers, wider Internet connections, and the like) and not also as a better design of web applications only based on software engineering issues, but it has to mainly be understood as student's successfulness in the same way the success of every business is measured by its customer's satisfaction. Therefore, the enhancement of the e-learning processes has to be developed from a starting point including as much student's information as it is possible (as every feature can potentially affect the success of a student in one way or another): from some of their personal features (sex, age, etc.) to their previous subject's qualifications, and also including their web navigation preferences and patterns, and others. Because of the need of analizing such potentially amount of information, data mining techniques are appropriate to apply here.

Data mining is also known as *knowledge discovery on databases* (KDD) and is a branch of the most generic data analysis field, which its main goal is to extract knowledge from the vast amount of data (large data sets) contained in every kind of databases. As stated in [3], the general experimental procedure adapted to a typical data mining process involves the following phases: *state the problem and formulate the hypothesis*, *collect the data*, pre-process *the data*, *estimate the model* and, finally, *interpret the model and draw conclusions*.

In this paper we show a case of study in the e-learning domain, for which we apply the different mentioned phases in a data mining process, from starting data sets that mainly include students qualifications for every learning activity susceptible to be qualified (as continuous evaluation works, practicums, etc.) with the main goal to achieve any conclusions that allow us to improve the e-learning process in some way.

In section *2 Data Mining in e-learning*, we state the goals for our case of study, which includes two domain problems to be modelled with data mining techniques. As mentioned, this is the theoretical starting point for the data mining process, but its not the only starting point from a practical point of view, because of the need of choosing the proper computer software tools that implement the data mining algorithms to be applied. This is not a trivial issue due to the vast variety of such tools and applications, and because every domain-related problem has its own characteristics and not every tool fits on them properly.

In this section we also discuss the data mining techniques and algorithms to be applied and how, it is interesting too, to obtain some conclusions regarding their effectiveness in our case of study. Additionally to these mentioned goals, we also propose to investigate how preprocessing the student qualifications (as a part of the e-learning data to be mined) by means of injecting new information on them (details are explained later) can affect the quality of the results.

Next, in section *3 Free Software for Data Mining*, we introduce a brief review of the current state of such software, because despite of there is quite a lot of suitable data mining privative software, the free software has many advantages for researchers as it can be freely adapted to the study case particularities or as it can be used at no cost. We also introduce here the free data mining software and tools we decided to use or develop for our case of study.

Once we have both stated the domain-related problems and the free software tools to carry out with the data mining, then in section *4 Running the Case of study*, we show which are the data sets we have to extract knowledge from, how this data were pre-processed in a e-learning basis, which are and how the experiments to carry out the model were conducted, and finally we expose the main results in a summarized manner.

But the results are meaningless if we are not capable to successfully reach the latest phase of the data mining overall process: to draw conclusions. In section *5 Conclusions* the most relevant conclusions are drawn and related with the initial domain-related problems and the other proposed goals.

## 2- Data mining in e-learning

Data mining applied to the e-learning domain is by itself a wide and emerging field in which a lot of research has been done and is continue doing nowadays. Very good summaries of the state-of-the-art in this discipline ranging its first ten years of life (from 1995 to 2005) can be found in literature [27] [28] as well as highly recommended books with representative compilations in this field [40].

E-learning is a global concept which entails the inclusion of every online learning, training and educational systems, which is referred as a new form of knowledge delivery through the advent and evolution of the World Wide Web. Thus, motivated by the web implementation of such e-learning systems and platforms, web mining is commonly used in the e-learning domain. Web mining is the application of data mining techniques to extract knowledge from web data, including web documents, hyper-links between documents, usage logs of web sites, etc. ([6]), and it has already been extensively applied to others domains such as e-commerce.

But, as in the e-commerce field customers are more willingness to report reviews, vote the products and participate in an active way (known as the intrusive way of collecting data from users), users (students and teachers) in the e-learning domain are usually not interested in such way, thus in this domain we have to obtain the user's data mainly from a non-intruse way. This not means, for instance, that quizzes can not be proposed to students to gain extra additional information, but their participation could not be as expected.

But web mining, with or without additional intrusive techniques, are mainly aimed to enhance the e-learning platform itself, with the improvement of the e-learning process as a consequence. For instance, the e-commerce sites commonly incorporate very powerful recommender systems, which means the web dynamically adapts to the user's profile. A lot of research has been done in developing similar recommender systems for e-learning platforms ([4]), or in developing adaptive (educational) hypermedia web systems (for instance, Interbook, ELM-ART, AHA! [15]) as another way to surpass the limitations of the static LMSs (*Learning Management Systems*) as Moddle, Atutor, etc. (free software) or WebCT, Virtual-U, etc. (privative software) amongst other examples. We should realize here that from static LMSs we can mainly address domain problems related with some kind of domain predictors or domain hypotheses as, for instance ([16]) it is the problem of predicting a student's final performance based on his or her work (submitting activities, forum participation, etc.) in a VLE (virtual learning environment).

There are also e-learning institutions that instead of one of those mentioned LMSs or adaptive web systems, use some kind of tailored e-learning platform adapted to its own designed learning planning and attending other issues as those related with marketing interests. A very known example of those is the UOC virtual campus [13], which is also designed following some recognized educational standards for e-learning, as SCORM [14], and has its own research lines for its improvement, as research in web mining usage that is already carried out [9], and more is currently in progress.

## 2.1- The e-learning domain-related starting points

Our case of study is intended, though, to be as general as possible (in the sense that its conclusions can be equally useful for every kind of those mentioned e-learning platforms -static LMSs, adaptive web systems and tailored virtual campus), thus we should address domain problems related with some kind of domain predictors or domain hypothesis as it will be the problem of predicting the student's final performance based on his or her previous qualifications (from submitting continous evaluation's activities to be assesed). This way, our approach does not offer us an improvement of the e-learning overall process througt an enhancement of the web learning platform, but from establishing predictors that allows us to group the students by their expected performance, as long as it is proved that heteregenous groups of students (including in the same work group students who are expected to have dissimilar performances) improves the overall successfulness for the full set of students.

*Predictors for the student's successfulness*

One direct way to measure the student's performance is to predict its final successfulness based on his or her previous qualifications. These can be referenced to qualifications from previous academic periods (academic semesters) or to qualifications he or she is achieving in the current academic semester for the different activities to be delivered and assesed as part of an evaluation methodology known as CE (Continuous Evaluation). We decide to use the later schema, as we piorize finding these predictors in relation with the characteristics of every subject over the student's trajectory througt different subjects.

*Predictors for student's rate of abandonment*

Another direct way to measure the student's performance is trying to predict whether he or she is going to leave the subject or not. Here we also make use of the CE qualifications in the current academic semester for the same reason, and as for the former mentioned domain-related problem, these qualifications can always be combined not only between themselves but also with some other student's features (as the age, if he or she is taking the subject by first time or not, etc.).

### 2.2- The Data Mining techniques to be applied

The data sets we use in our case of study (see section *4 Case of study*) include either the final subjects' qualifications for every student and a way to derive a dichotomized attribute telling us whether the student has left the subject or not. In consequence, there is no need for a training data set because we already have what the predictors should match. This lead us to the possibility of using an unsupervised data mining technique.

Without excluding other unsupervised data mining techniques in future lines of our research, we have chosen the following well-known clustering methods, as we think it is a good starting point (only a brief description is offered here, please see references for further information):

- Simple K-means

This is a partitional clustering method, where we previously must specify the number of K clusters we want to obtain. We also have to choose what points are considered as the initial centroids for every cluster (different chosen initial centroids usually bring up different results, so this algorithm is sensitive to this parameter). In every step each observation is assigned to the cluster with the nearest centroid and once all assignments are done, the centroids are recalculated.

- Hierarchical clustering

This is an agglomerative algorithm, meaning it starts from as many clusters as observations or samples we have and, step by step, these observations are being grouped forming bigger and bigger clusters. The criteria to decide which observations join one cluster or another is based in two parameters: a distance, or dissimilarity function, and the type of linkage between clusters. With this algorithm there is no need to specify the number of resulting clusters as a hierarchical clustering is created where we can choose from the number of desired clusters.

In [41] you can find a detailed description of the hierarchical clustering and a simple variant of the K-means methods, and a comparison between them.

- Expectation – Maximization (EM)

This algorithm is based on the same principle as the K-means, but instead of maximizing the distances between clusters, it maximizes the probability for all observations of belonging to each cluster, according to one or more probability distributions (it depends on the specific implementation). For a detailed description see [42].

One issue we may not forget is to consider the time complexity of such algorithms, because this characteristic can make an algorithm impossible to apply in certain real conditions when we need to extract knowledge from large data sets. As it is mentioned in the references [41] and [43], K-means and EM have a linear time complexity, while the hierarchical clustering algorithm has a quadratic one. Knowing this, we did not measure the time cost for every algorithm used in our case of study.

The reasons we've chosen three algorithms instead of one are:

– it is more plausible, if not sure, to obtain more confident conclusions from results based on a three algorithm basis instead from one;

– it is also interesting to check the effectiveness of such algorithms for our formulated domain-related problems and find out which of them fits better for the different data.

## 2.3- Injecting domain-related knowledge to SQ

Student qualifications (SQ) are usually given as nominals, this means their problem resides in keeping no ordinal information, thus the distance between an A (the highest SQ) and a B (the next SQ in the rank, from the top) is exactly the same as the one between that A and a D (the lowest SQ).

| *Nominal Student qualifications in a downward order* | | |
|---|---|---|
| Nominal | Mean | State |
| A | Excellent qualification | Successful |
| B | Notable qualification | Successful |
| Cm | Sufficient qualification | Successful |
| Cn | Insufficient qualification | Unsuccessful |
| D | Very Insufficient qualif. | Unsuccessful |
| NP | Not presented | Unsuccessful |

As the data mining software usually, if not always, treat nominals with a unitary distance when are different, and with a zero distance when are equal, regardless of whether the SQ are higher or lower in the rank, when clustering algorithms are applied to these data there is no way for them to know and to take into account their ordinality, because the ordinality of nominal SQ is a prior domain knowledge and it is not included with SQ themselves, unless these SQ were given as numbers instead of nominals.

Encoding from nominal to ordinal (numeric) can be done in different ways, both automatically and manually. When the encoding process is automatic, it is based on some statistical method (for instance, assigning frequency values to the nominals or dichotomizing them), but these techniques have no sense for SQ. Instead, we should use manual encoding to fully add our domain prior knowledge to the nominal data. For example, we could use the following default numeric qualifications:

| *Default nominal to ordinal numeric encoding* | |
|---|---|
| Nominal | Ordinal numeric |
| A | 9.5 |
| B | 7.5 |
| Cm | 5.5 |
| Cn | 4.5 |
| D | 2.5 |
| NP | 0 |

And this replacement can only be done manually by a domain user who has this knowledge.

Once we have the ordinal information within the SQ themselves (both from a nominal to numeric manual replacement or from SQ originally offered as numeric values) then the data still has a lack of information: they do not contain anything regarding whether each SQ is a successful or an unsuccessful one. This information is only known by the domain user (who knows that from a threshold SQ value to up the qualifications are successful, and to down are unsuccessful) but the data themselves can not say anything about that. This also means the data mining algorithms and techniques applied to these data can not take into account any information related with the successfulness feature of a SQ value, thus the obtained results could be somehow biased. This is specially relevant when conclusions has to be achieved through contrasting the results with some successfulness pattern, as it is in our case study.

When the used data mining techniques are based on similarity functions (or dissimilarity functions, as a wide range of distance functions) we can hypothesize that one way for adding, or injecting, successfulness information to the data to be mined is by emphasizing somehow the distances between successful and unsuccessful student qualifications.

*The klog function*

A non-parametric method to do this is by adding a dichotomized attribute to the key attributes (the key attributes are the ones to be data mined). This new attribute has one unique value for each successful SQ sample, and another for those that are unsuccessful. A simple example is given next.

Suppose we have to cluster SQ in successful and unsuccessful. If we only have the ordinal numeric attribute, then the dissimilarity function (euclidean distance here) gives us:

$$d_1 = \sqrt{(7.5-5.5)^2} = 2$$
$$d_2 = \sqrt{(5.5-4.5)^2} = 1$$

As it is shown, with no additional information, the distance between two successful SQ can be even greater than between a successful and an unsuccessful.

But if we add the dichotomized attribute, then we have:

$$d_1 = \sqrt{(7.5-5.5)^2+(1-1)^2} = 2$$
$$d_2 = \sqrt{(5.5-4.5)^2+(1-0)^2} = \sqrt{2} \; ; \; 1.41$$

Thus, adding a dichotomized attribute we can provide data with useful but limited information, because there is no way to make $d_2$ greater than $d_1$ . To do that we need a parametric method, as it is provided by the *klog* function.

Through the *klog* math function we can also emphasize the dissimilarity between successful and unsuccessful SQ, but in a no-limited manner. This is it because the strength of the *klog* effect that it has on the data depends on the value of a parameter called *'factor k'* or simply k, and this parameter can be freely adjusted by the investigator. The *klog* function is defined as:

$$\mathrm{klog}(v_i) = \left\{ \begin{array}{ll} v_i \cdot k^{\frac{\ln(\mathrm{klog}(v_{i-1})) - \ln v_i}{\ln(2k)}} & \text{if } v_i \geq v_{Threshold} \\ v_i \cdot k^{\frac{-\ln v_i}{\ln(2k)}} & \text{if } v_i < v_{Threshold} \end{array} \right\} \forall v_i, k \geq 1$$

where

$v_i = i$-th SQ (student qualification) from the SQ top-down ordered sequence of different possible qualifications.

$v_{Threshold}$ = threshold value between successful and unsuccessful SQ

As it is shown, *klog* is an iterative function over a top-down ordered set of SQ, where i=1,...,n with n the number of qualifications (n=6 for the nominal qualifications shown above). As long as the threshold value is not surpassed (the Cm qualification in the example), the behaviour is that all values tend to reach its owns upper value as much as *k*'s value increases:

$$\lim_{k\to\infty}\left[\mathrm{klog}(v_i)\right] = v_i \lim_{k\to\infty}\left[k^{\frac{\ln v_{i-1} - \ln v_i}{\ln(2k)}}\right] = v_i \cdot e^{\ln v_{i-1} - \ln v_i} = v_i \cdot \frac{v_{i-1}}{v_i} = v_{i-1}$$

When the threshold value has been surpassed the behaviour is not trying to reach the one's lower value but the zero. This is a slightly different behaviour due to the nature of the top-down iteration. It is always possible to apply the first formula to both intervals if the iteration for unsuccessful qualifications is done in a down-top order.

Therefore, increasing the value of k, with the *klog* function it is possible to emphasize the distance between successful and unsuccessful qualifications as much as it is desired. For *k = 1* (its lowest possible value) the numeric qualifications are not transformed at all, while with *k* tending to infinite, the numerical SQ attribute converges to a dichotomized attribute.

| *Injecting successfulness prior domain knowledge to Student Qualifications with the klog function* | | | | | | |
|---|---|---|---|---|---|---|
| Nominal | Numeric replacement | Normalized [0,1] (k = 1.0) | k = 2.0 | k = 100.0 | k = infinite | State |
| A | 9.5 | 1 | 1 | 1 | 1 | Successful |
| B | 7.5 | 0.79 | 0.89 | 0.97 | 1 | Successful |
| Cm | 5.5 | 0.58 | 0.73 | 0.91 | 1 | Successful |
| Cn | 4.5 | 0.47 | 0.21 | 0.05 | 0 | Unsuccessful |
| D | 2.5 | 0.26 | 0.12 | 0.03 | 0 | Unsuccessful |
| NP | 0 | 0 | 0 | 0 | 0 | Unsuccessful |

Because of this condition:

$$v_i \geq 1; \forall i = 0 \pm n$$

before applying the *klog* function values must be normalized upper from 1, thus the steps for injecting prior domain-related knowledge to student qualifications are:

1. Numeric manual replacement, if qualifications are given as nominals;
2. Normalization in [1,2];
3. Transformation through the *klog* iterative function with the desired k value.
4. Normalization in [0,1].

For normalization we use the next transformation, also applied in a top-down ordered sequence of student qualifications:

$$N_{a,b}(v_i) = \left\{ \begin{matrix} N_{a,b}(v_{i-1}) - (b-a)\dfrac{v_{i-1} - v_i}{\max - \min} & \text{if } i > 0 \\ b & i = 0 \end{matrix} \right\} \forall i = 0 \pm n$$

where *a,b* are the lower and upper quotas for the interval of normalization, and *min, max* are the minimum and maximum values from the set of qualifications to be normalized.

### 2.4- Summary of the goals

As it is explained, our case of study mainly has the following data-mining in e-learning goals:

**1-** To find out if there are students' qualifications from CE (Continuous Evaluation) to be considered as a good candidates, individually or in combination between themselves or with other type of students' features, for predicting:

– the student's successfulness;
– and the student's abandonment rate.

**2-** To compare the three selected clustering algorithms and determine which best fit to our domain-related problems:

– Simple K-means
– Hierarchical clustering
– Expectation-Maximization

**3-** To determine if the injection of prior domain knowledge to the data can improve the results or whether there is a value of the *k factor* for which the results are optimal.

## 3- Free software for Data Mining

### 3.1- A brief review

Free software has many advantages over the proprietary. It is demonstrated in [44] that free software has risen to great prominence due to its multiple advantages. Besides its potentially superior quality, the advantages reside on its licenses, which allow users the freedom to run the software for any purpose, to study and freely modify the program, and redistributing copies of either the original or the modified program, without having to pay any royalties to previous developers. Therefore, the free software can be used at no cost, adapted or modified to the researcher's specific-domain needs, and what is also important, it makes you no longer dependent on any software company.

Nowadays, free software is also been widely used and developed by investigators in their data mining research projects. For instance, in [45] (Weka), [46] (Weka), [47] (Tanagra), [48] (Tanagra), [49] (RapidMiner) and so many others, free tools are used to conduct the data mining experiments. But the researchers not only use existing free data-mining tools, we also modify or develop our own tools in many cases. For instance, the EPRules tool ([25] and [32]) is a good example of a free software tool specifically developed by the own researchers to address the corresponding particular study cases of their researches. EPRules is oriented to be used by non-experts in data mining, and to discover knowledge from SHAEW (*"Sistemas Hipermedia Adaptativos para la Educación basada en Web"*) adaptive web systems, in the form of prediction rules.

After a search on the Internet for free data mining software, we discovered the existence of many tools, applications and suites developed throughout the scientific community. It is recommended to visit the *KDnuggets* [2] and *The-Data-Mine* [50] sites for an overall view of them and the proprietary data mining software, too. Both are complete Data Mining resources, but with KDnuggets we must be careful when looking for free software solutions here, because in this web site the free software is not always clearly distinguished from the freeware software. Instead, in The-Data-Mine site you can easily pick up free / open source software by entering the free or open source keywords to its searching textfield in the software section. From all of the data-mining free software found on the Internet, and similarly as it is reported in [51], we can highlight the following tools due to their relevance to the scientific community:

– Weka [37]: Weka is the most well-known free software tool for data mining and machine learning. Its algorithms can either be applied directly to a dataset from its own interface or used in your own Java code. Weka contains tools for data pre-processing, classification, clustering, association rules and visualization. We make use of its clustering capabilities to carry out our study case, because it implements the three chosen clustering algorithms. Weka is also chosen because it is, afterall, the leader in the data-mining free software world, as it is stated when the most part of the other free tools are Weka-compatible, and when it is used in the majority of research data mining projects.

– Tanagra [38]: This tool includes the majority of the data mining techniques, as statistical and automatic learning, supervised algorithms, clustering, parametric and non-parametric statistics, association rules, decision trees, and so on. It is specifically designed for academic purposes and thus, it does not include commercial features as connection with data warehouses, graphical interactivity and the like. Instead, it is compatible with Weka.

– MiningMart [52]: MiningMart is developed with the purpose of re-using best-practice cases of pre-processing large and very large data sets. Thus, this tool is not focused on the overall data-mining process but only in one of the first steps, the data pre-processing, which is usually the most time-consuming and the one that has the greatest influence on the quality of the results.

– KNIME [53], [54]: This is a modular environment intended to enable easy integration of new algorithms, data manipulation and visualization methods as models. It is also compatible with Weka and moreover it includes statistical methods via an embedded usage of R ([1]).

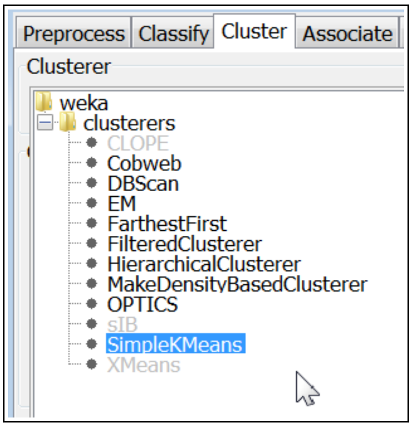
– ADaM [55]: This free software is an integrated toolkit packaged as a suite of independent components intended to be used in grid or cluster environments. It provides feature selection capabilities, image processing and data cleansing. It is also Weka compatible.

– RapidMiner (formerly known as Yale [56]): Yet another free and open source environment for Knowledge Discovery and Machine Learning. It is highlighted as it provides a rich variety of methods which allow the prototyping of new applications and also makes costly re-implementations unnecessary. It also supports the Weka native standard file format (ARFF).

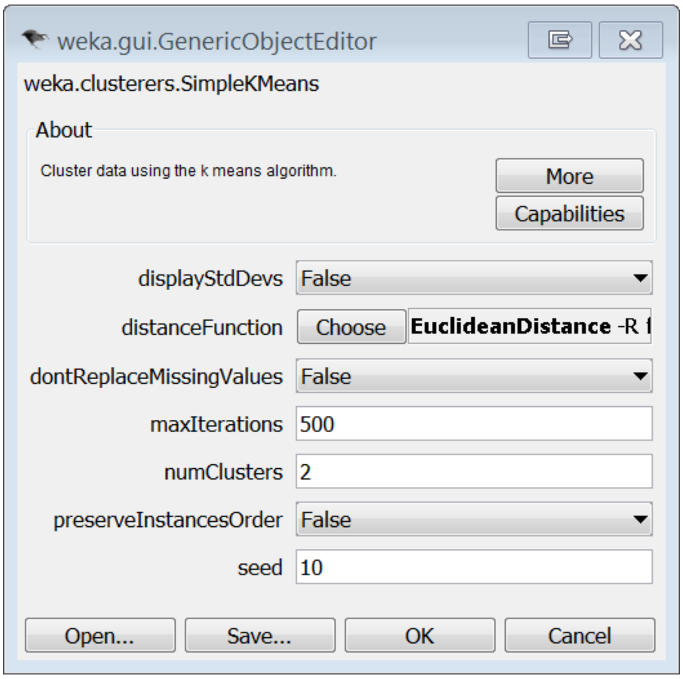
## 3.2- Weka for pre-processing and clustering

*Clustering*

In the next screen cap the clustering algorithms supported by Weka are shown:



The configuration options for Weka's implementation of the three chosen clustering algorithms are described next. For the Simple K-means, we have:



weka.clusterers.SimpleKMeans

About

Cluster data using the k means algorithm.

displayStdDevs: False

distanceFunction: EuclideanDistance -R f

dontReplaceMissingValues: False

maxIterations: 500

numClusters: 2

preserveInstancesOrder: False

seed: 10

Where the relevant options are:

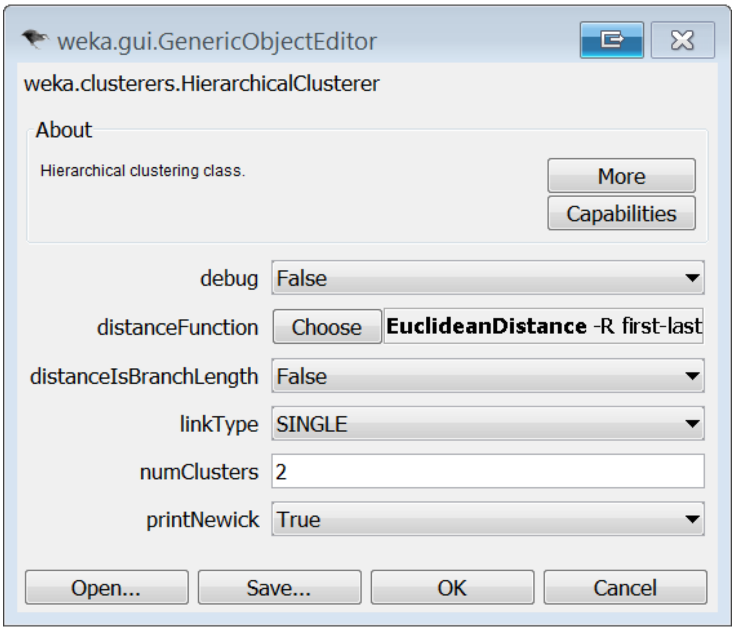
– Distance Function: can be either the Euclidean distance, the Manhattan distance, also the Chebyshev, the Edit, or the Minkowski distance (it is a lack not offering others as those based on correlations). If the Manhattan is used, then centroids are computed as the component-wise median rather than the mean. For our case of study we used the default euclidean distance.

– Number of Clusters: the number of clusters in which the data have to be partitioned by the algorithm. In our case of study, as the stated domain-related problems are dichotomic (success / unsuccess and abandonment / not abandonment) we always needed to partition the data in two clusters.

– Seed: this is the random number seed to be used by the Weka implementation of the K-means to calculate the initial centroids. Note that, in general, K-means is quite sensitive to how clusters are initially assigned, thus to the value of the seed parameter. As we'll see later, we tried different seed values and kept the optimal comparing the results obtained.

– Other considerations: Weka Simple K-means algorithm automatically handles a mixture of categorical and numerical attributes. Furthermore, the algorithm automatically normalizes numerical attributes we doing distance computations, where distances between categorical are assigned to 1 when are categorical values are different and to 0 when equal.

For the hierarchical clustering, additionally to the number of clusters and the distance function used (here we can choose the same distances mentioned above, as are programmed as an independent module) another key configuration parameter is the link type. The link type is used to establish how the distances between clusters are calculated, besides the distance function used. For instance, with the Single link the inter-clustering distance is measured as the minimum distance between every pair of members from each cluster, while with the Complete link, the maximum distance is used. Weka allows us to choose between a variety of link types (Single, Average, Complete, Mean, Centroid and others) from which we pick up the Single and the Mean linkages just to compare their performances in terms of quality of the results.


weka.gui.GenericObjectEditor

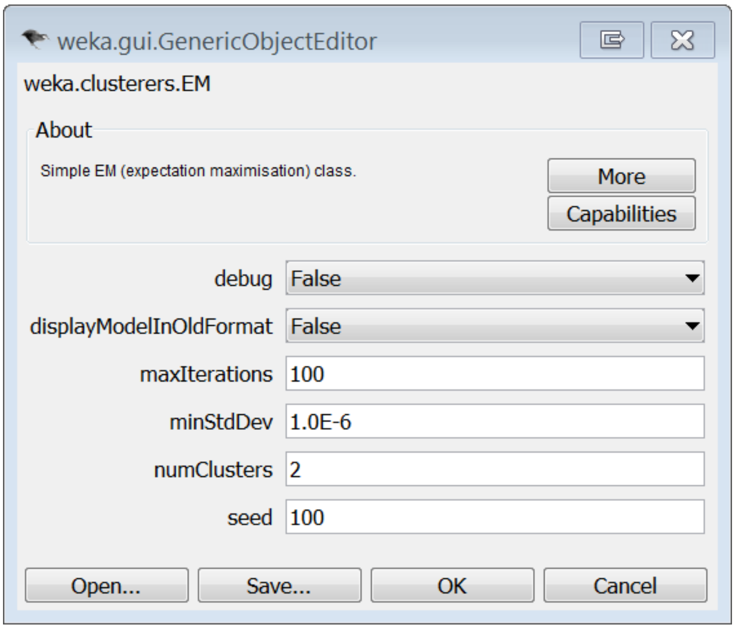
weka.clusterers.HierarchicalClusterer

About

Hierarchical clustering class.

| | |
|---|---|
| debug | False |
| distanceFunction | EuclideanDistance -R first-last |
| distanceIsBranchLength | False |
| linkType | SINGLE |
| numClusters | 2 |
| printNewick | True |

Regarding the E-M algorithm, as it is described above, it is based in the same partitioning principle as the K-means, but using probability distributions instead of distances, taking the view that while every instance belongs to only one cluster, it is almost impossible to know for certain to which one. Thus, the basic idea is to approximate every attribute with a statistical finite combination of probability distributions. The Weka implementation uses a simple combination of Gaussian distributions, each of them with different mean and variance values that has to be calculated based on the input data. From the options dialog we can chose the following parameters:



weka.clusterers.EM

Simple EM (expectation maximisation) class.

debug False
displayModelInOldFormat False
maxIterations 100
minStdDev 1.0E-6
numClusters 2
seed 100

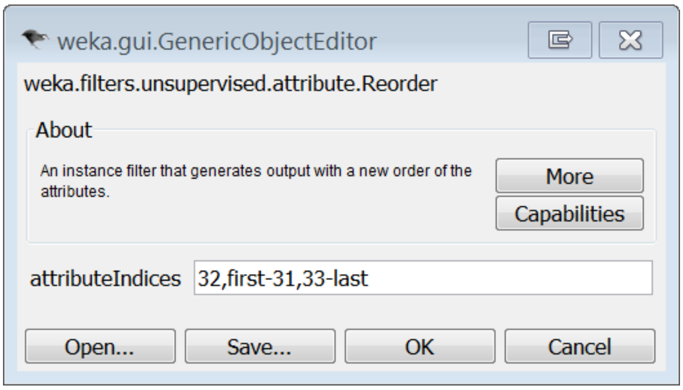
Again, the number of clusters is left to 2, although if it is set to the default (-1) then the algorithm automatically determine the optimal number of clusters using a 10-fold cross-validation. And the seed were also tested in our case of study to establish its optimum value for our data sets.

*Preprocessing*

Weka allows us to do some editing operations on the instances, besides from applying over them an extensive collection of preprocessing filters. The preprocessing filters used in our case of study are:

```
unsupervised.attribute.Reorder
```

This filter is for reordering attributes. This is not indispensable at all, but reordering attributes can make the researchers experimentation tasks easier, as the attributes are always found in the same order. For instance, the next reorder the attribute sorted in the 32th position as the first:



weka.gui.GenericObjectEditor

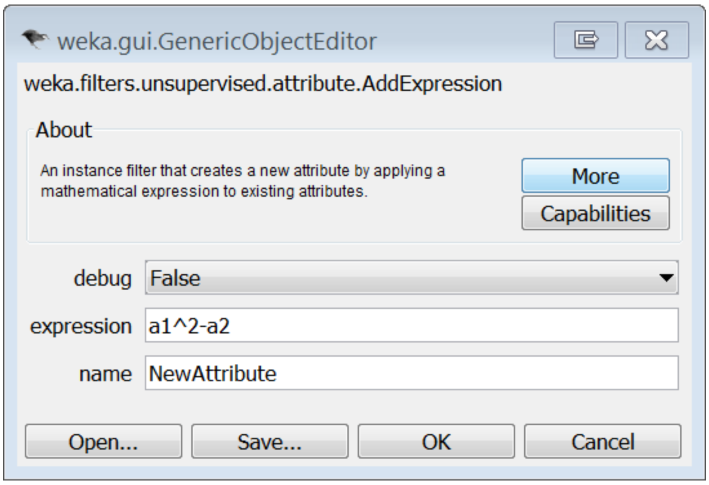
weka.filters.unsupervised.attribute.Reorder

About

An instance filter that generates output with a new order of the attributes.

attributeIndices: 32,first-31,33-last

This one is intended for creating a derivative new attribute based on some mathematical relation applied to numeric instances from one attribute or between two or more attributes. As an example, the following creates a new attribute with the result of applying a mathematical relation between the numerical attributes 1 and 2:



weka.gui.GenericObjectEditor

weka.filters.unsupervised.attribute.AddExpression

About

An instance filter that creates a new attribute by applying a mathematical expression to existing attributes.

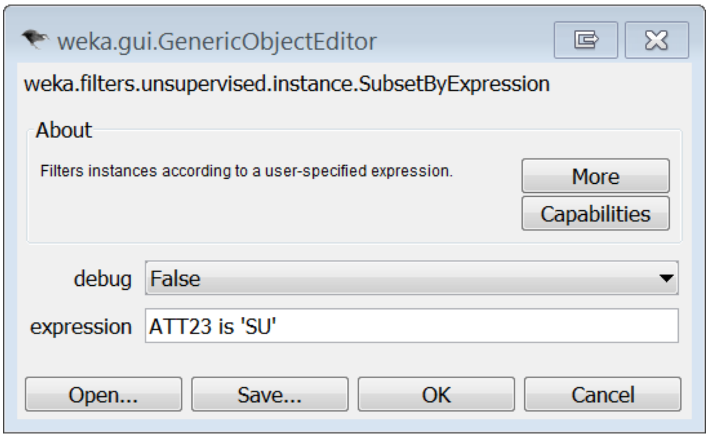
debug: False

expression: a1^2-a2

name: NewAttribute

`unsupervised.instance.SubSetByExpression`

With this filter is possible to partition the set of instances or observations into different subsets according to some condition. For instance, if we have a dichotomized attributed telling us whether a student has successfully passed the subjects' final assessment or not, this filter can partition the data set into two subsets: the first for the successful students and the second for the others. The next cap shows an example of how from the dichotomized attributed number 23 the subset with the 'SU' value is extracted.



weka.gui.GenericObjectEditor

weka.filters.unsupervised.instance.SubsetByExpression

About

Filters instances according to a user-specified expression.

debug False

expression ATT23 is 'SU'

*Other pre-processing tasks. Limitations.*

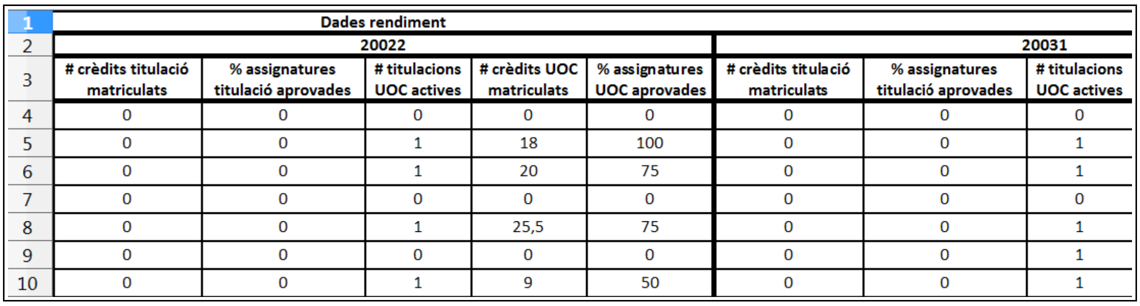
Manually editing multiple instance values, in the Edit table, is well supported for numeric values, but not for nominal (version 3.7). On the other hand, filters for discretizing or categorizing numeric attributes only support automatic ways based on some statistical calculated values, and it is also limited in the sense that the researcher can not proceed with entering customized nominal values (unless done one by one in the Edit table) and manually establishing the interval limits.

### 3.3- LiWeCool for preprocessing datasets

LiWeCool [57] (LIght Weight WEka COmpatible data preprocessing toOL) tool is the free software, under the GPL v3, developed by the author of this paper, with the goal to automatize the specific preprocessing tasks needed for our case of study and that are not found on any of the other free software tools, as the ones mentioned in section 3.1.

*Making CSV file Weka-compatible*

When datasets are given as spreadsheet files, they often have the attributes' information spread over various rows instead of one.

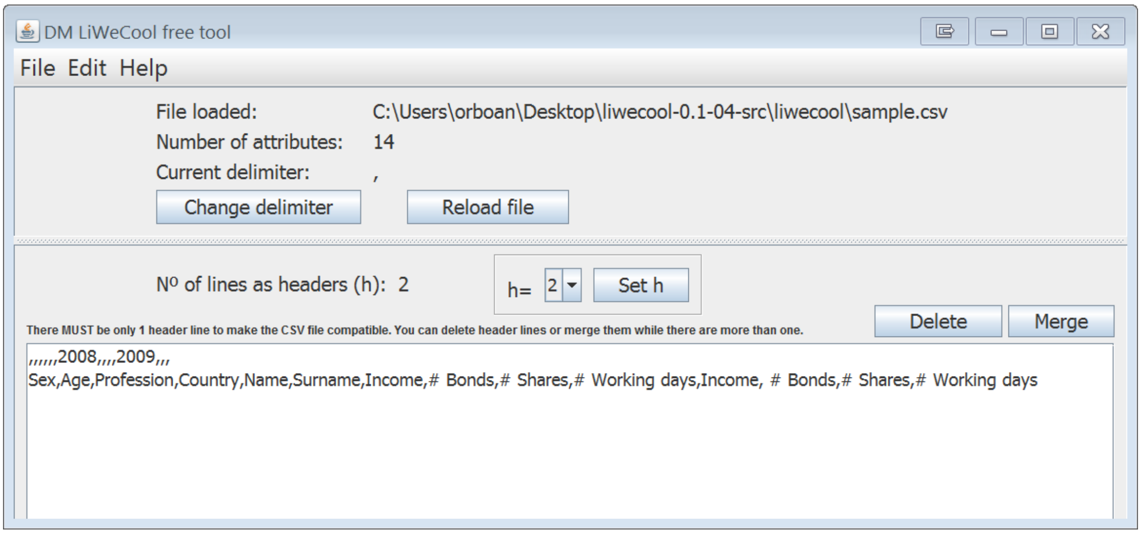


| 1 | Dades rendiment | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2 | 20022 | | | | | 20031 | | |
| 3 | # crèdits titulació matriculats | % assignatures titulació aprovades | # titulacions UOC actives | # crèdits UOC matriculats | % assignatures UOC aprovades | # crèdits titulació matriculats | % assignatures titulació aprovades | # titulacions UOC actives |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 1 | 18 | 100 | 0 | 0 | 1 |
| 6 | 0 | 0 | 1 | 20 | 75 | 0 | 0 | 1 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 1 | 25,5 | 75 | 0 | 0 | 1 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 10 | 0 | 0 | 1 | 9 | 50 | 0 | 0 | 1 |

As Weka can not load spreadsheet files, the datasets has to be exported as a Weka-compatible format. CSV is a common file format admitted either by spreadsheet applications and by Weka, but when a spreadsheet dataset with various header rows is exported as CSV, this CSV can not be loaded by Weka, because Weka only supports CSV files with a unique first line with the header role (thus, all attributes' information has to be only stored in the very first line).

LiWeCool can deal with such problem by loading multiple header CSVs and converting them as Weka-compatible CSV. The user has two options: to remove the header lines with redundant or not relevant information, or to merge the header lines resulting in one containing the information from others.

Before merging:



File loaded: C:\Users\orboan\Desktop\liwecool-0.1-04-src\liwecool\sample.csv

Number of attributes: 14

Current delimiter: ,

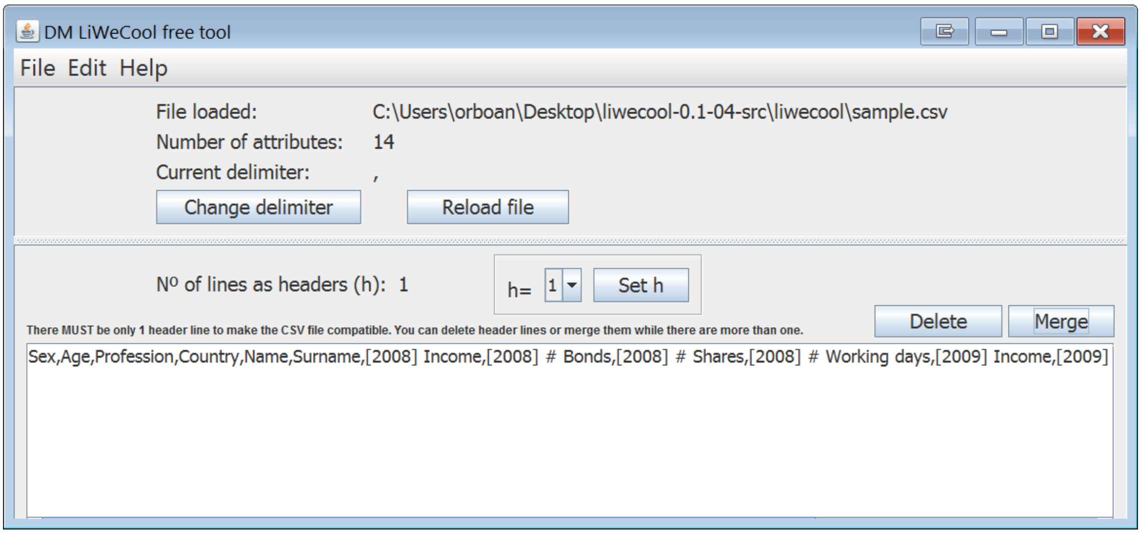
Nº of lines as headers (h): 2

h= 2

There MUST be only 1 header line to make the CSV file compatible. You can delete header lines or merge them while there are more than one.

```
,,,,,,2008,,,,2009,,,
Sex,Age,Profession,Country,Name,Surname,Income,# Bonds,# Shares,# Working days,Income, # Bonds,# Shares,# Working days
```

After merging:



File loaded: C:\Users\orboan\Desktop\liwecool-0.1-04-src\liwecool\sample.csv

Number of attributes: 14

Current delimiter: ,

Nº of lines as headers (h): 1

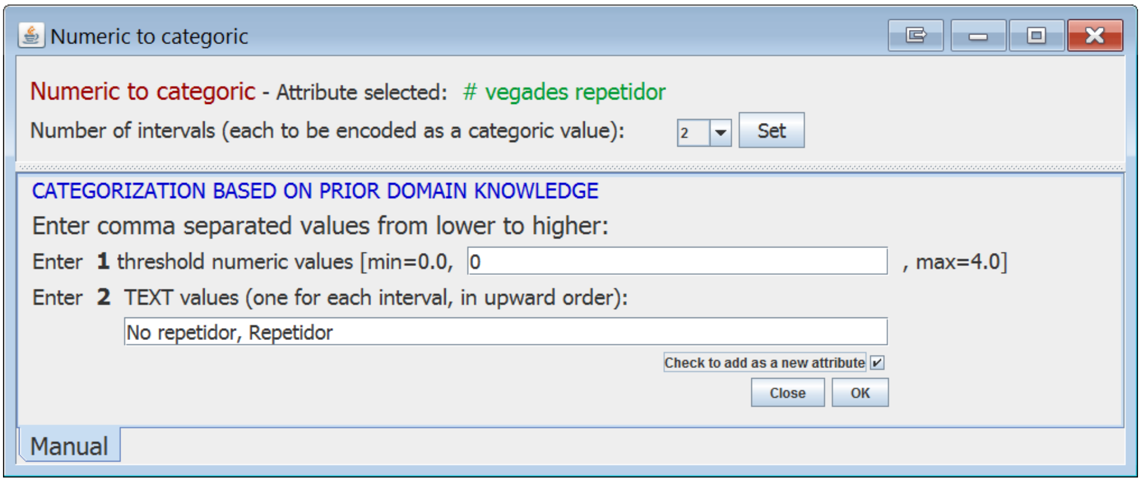
h= 1

There MUST be only 1 header line to make the CSV file compatible. You can delete header lines or merge them while there are more than one.

```
Sex,Age,Profession,Country,Name,Surname,[2008] Income,[2008] # Bonds,[2008] # Shares,[2008] # Working days,[2009] Income,[2009]
```

*Categorization based on prior domain knowledge*

This feature is intended to transform numeric intervals into a nominal, but allowing the researcher to manually establish the number of intervals, their quotas, and entering customized strings as nominals, all based on the prior domain knowledge the researcher has. The following example shows how a numeric attribute called '# vegades repetidor' (the number of times a student re-takes the subject as he or she failed to pass it) can be categorized into a dichotomic attribute with the values {'No repetidor', 'Repetidor'}:



Numeric to categoric - Attribute selected: # vegades repetidor

Number of intervals (each to be encoded as a categoric value): 2

CATEGORIZATION BASED ON PRIOR DOMAIN KNOWLEDGE

Enter comma separated values from lower to higher:

Enter 1 threshold numeric values [min=0.0, 0 , max=4.0]

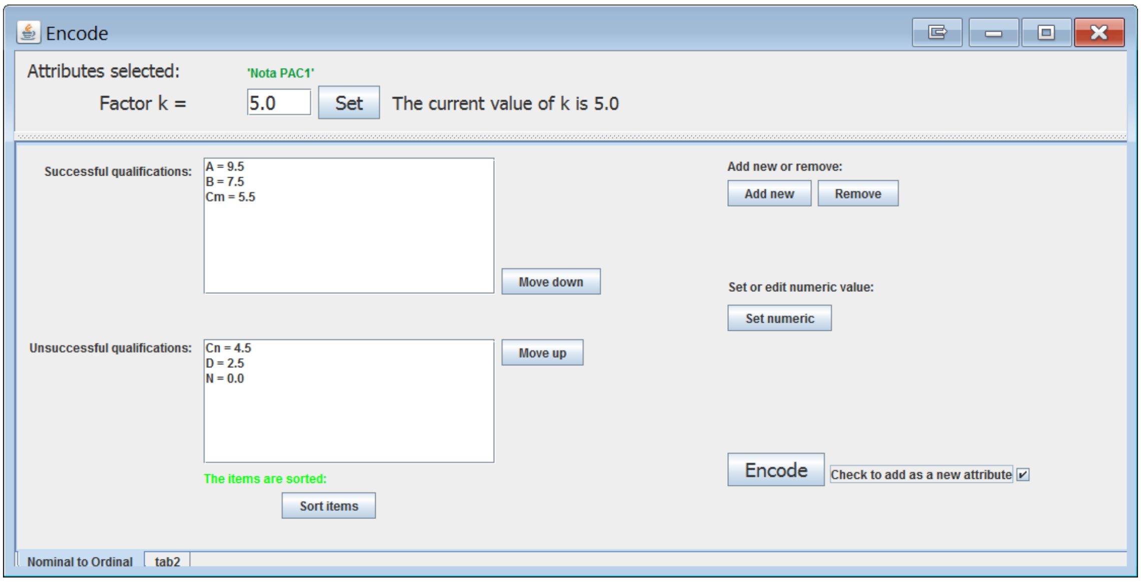
Enter 2 TEXT values (one for each interval, in upward order):

No repetidor, Repetidor

Check to add as a new attribute ☑

*Encode Student Qualifications*

As explained in section *2.3- Injecting domain-related knowledge to SQ*, there is a goal in testing whether the results are improved or not with providing ordinality to the nominal qualifications and, furthermore, transforming those numeric qualifications through the *klog* function as a way to emphasize the dissimilarity between successful and unsuccessful qualifications. The possibility of improving the results by injecting successfulness information with the *klog* function transformation is an hypothesis proposed here, thus we did not expect it was supported by any existing free tool. Therefore, this feature has been implemented in LiWeCool.



Attributes selected: 'Nota PAC1'

Factor k = 5.0

The current value of k is 5.0

Successful qualifications:
A = 9.5
B = 7.5
Cm = 5.5

Unsuccessful qualifications:
Cn = 4.5
D = 2.5
N = 0.0

The items are sorted:

Check to add as a new attribute ☑

In this window frame, the user can add or remove nominal student qualifications, because the loaded nominals are the ones included in the selected attribute and the selected attribute may not contain all the possible qualifications values. LiWeCool assigns default numeric values for the nominal qualifications used by the UOC e-learning university, but the user is free to modify these defaults. As it is shown, the user can also enter the desired value of k for the klog transformation, which is applied by means of establishing the threshold value between successful and unsuccessful qualifications, moving them up and down between the two boxes.

*Other features*

LiWeCool can also normalize numeric qualifications based on the klog transformation and make multiple value replacements for the different attributes selected, both for nominals and numeric values. Besides creating CSV Weka-compatible files, it directly supports loading and saving datasets in the ARFF (Weka native) format.

## 4- Running the Case of study

### 4.1- The data

The datasets available for our case of study are in the form of spreadsheet files (XLS but saved as ODS with OpenOffice). Each file belongs to one subject:

– <u>Subject 1.ods</u>: This a UOC subject. It contains information belonging to two academic semesters with various types of student attributes, including:

- personal features: sex, year of birth;

- subject features: CE qualifications *as nominals*, and the number of times the student has repeated the subject;
- Access profile (when entering the University);
- and students' performance data for the last academic semesters from the 19971.

  The number of instances per semester are:
  - 20081: 43 instances.
  - 20082: 41 instances.

– <u>Subject 2.ods</u>: This is another UOC subject. It also contains the same attributes as for Subject 1, and the number of instances are here:

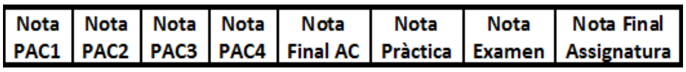
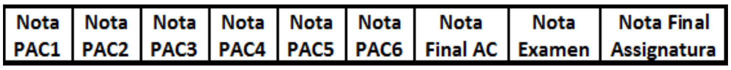
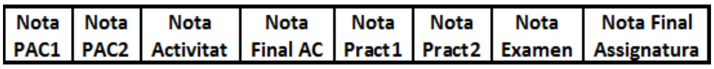
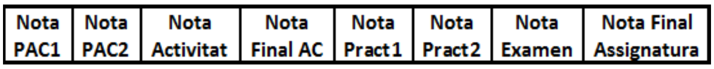
- 20081: 87 instances.
- 20082: 83 instances.

– <u>Subject 3.ods</u>: This is a non-UOC subject. Its data belongs to a unique and unknown academic semester, with 89 instances, and the number of attributes are quite reduced (in comparison with the UOC subjects):

- personal features: only sex is provided here;
- subject features: a dichotomized attribute telling us whether a student is repeating the subject or not, and the CE qualifications provided as numbers (in contrast with the nominals from the UOC subjects).

– <u>Subject 4.ods</u>: This is another non-UOC subject. The provided data is the same as for the subject 3, having also only one academic semester with 89 instances.

Besides these descriptions, the subjects are different subjects, ie. there is no reason why they should have the same CE structure. In fact, only the CE structures for the non-UOC subjects are equal between them, as it is shown next:

| **Subject** | **CE structure** |
|---|---|
| **1** (UOC) | <br>Nota PAC1<br>Nota PAC2<br>Nota PAC3<br>Nota PAC4<br>Nota Final AC<br>Nota Pràctica<br>Nota Examen<br>Nota Final Assignatura |
| **2** (UOC) | <br>Nota PAC1<br>Nota PAC2<br>Nota PAC3<br>Nota PAC4<br>Nota PAC5<br>Nota PAC6<br>Nota Final AC<br>Nota Examen<br>Nota Final Assignatura |
| **3** (non-UOC) | <br>Nota PAC1<br>Nota PAC2<br>Nota Activitat<br>Nota Final AC<br>Nota Pract1<br>Nota Pract2<br>Nota Examen<br>Nota Final Assignatura |
| **4** (non-UOC) | <br>Nota PAC1<br>Nota PAC2<br>Nota Activitat<br>Nota Final AC<br>Nota Pract1<br>Nota Pract2<br>Nota Examen<br>Nota Final Assignatura |

Our case of study worked out with the CE qualifications (nominals for the UOC subjects, and numeric for the non-UOC) and additionally with the dichotomized attribute of 'repeater student'. All these attributes are called 'key attributes', which means the subject attributes are the key attributes, ie, the ones from where we shall try to obtain the predictors. Any key attribute or any combination of key attributes can potentially be assessed as a predictor for one of the two stated domain-related problems or the other (predicting successfulness and abandonment rate). The rest of attributes will be assessed in future work.

### 4.2- Preprocessing the data

The following are the steps in creating *ready-to-Weka* files (ready to be clustered) from the spreadsheet data files:

Starting files are XLS spreadsheet OpenOffice-compatible files. After the number of each step, it is indicated between square brackets the free software used in that step.

1 – [OpenOffice Calc] Open XLS with OpenOffice Calc.

2 – [OpenOffice Calc] If numeric values are given in a comma decimal format, instead of a dot, then change locale settings.

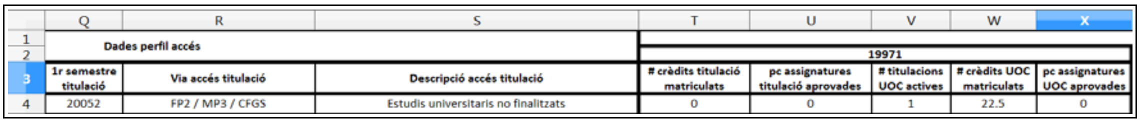
3- [OpenOffice Calc] Export from OpenOffice.Calc as a CSV file.

*Cleansing data:*

- Subjects 1 and 2: beware of some nominals containing the characters {,}, {'} or {%} because are used as a comma delimiter or conflict with Weka parsing rules:
  - {E.T. de Telecomunicació, especialitat Telemàtica} has been replaced by {E.T. de Telecomunicació - especialitat Telemàtica}
  - {CFGS Producció audiovisual, ràdio i espectacles}, by {CFGS Producció audiovisual - ràdio i espectacles}
  - In catalan, {'} is used as a sintactic char, thus some nominals include it. It's been replace by a simple white space.
  - {%} substituted by the acronim {pc}

4- [LiWeCool] Load CSV with LiWeCool and treat the header lines properly (so the CSV files will be Weka-compatible).

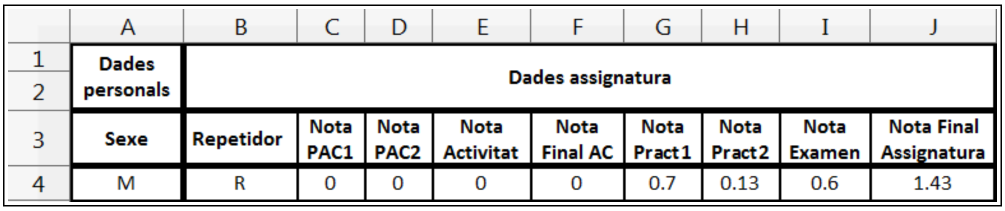
- Subjects 1 and 2: remove the first header line and merge the second with the third.



| | Q | R | S | T | U | V | W | X |
|---|---|---|---|---|---|---|---|---|
| 1 | Dades perfil accés | | | | | | | |
| 2 | | | | 19971 | | | | |
| 3 | 1r semestre titulació | Via accés titulació | Descripció accés titulació | # crèdits titulació matriculats | pc assignatures titulació aprovades | # titulacions UOC actives | # crèdits UOC matriculats | pc assignatures UOC aprovades |
| 4 | 20052 | FP2 / MP3 / CFGS | Estudis universitaris no finalitzats | 0 | 0 | 1 | 22.5 | 0 |

*Spreadsheet original data files for subjects 1 and 2 come with 3 headers*

- Subjects 3 and 4: delete the two first header lines.



| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Dades personals | Dades assignatura | | | | | | | | |
| 2 | | | | | | | | | | |
| 3 | Sexe | Repetidor | Nota PAC1 | Nota PAC2 | Nota Activitat | Nota Final AC | Nota Pract1 | Nota Pract2 | Nota Examen | Nota Final Assignatura |
| 4 | M | R | 0 | 0 | 0 | 0 | 0.7 | 0.13 | 0.6 | 1.43 |

*Spreadsheet original data files for subjects 1 and 2 also come with 3 row headers*

5- [Weka] Load Weva-compatible CSV files with Weka and proceed with some preprocessing tasks:

- Subjects 1 and 2 (five academic semesters):
  - Reordering attributes to match the same order as in subjects 3 and 4 (just to make the researcher's work easier) (Weka filter '`unsupervised.attribute.Reorder`')
  - Inferring the age from the year of birth and adding it as a new attribute (Weka filter '`unsupervised.attribute.AddExpression`').

5- [LiWeCool] *Cleansing data:*

- Subjects 3 and 4:
  - Cleansing the *'Repetidor'* attribute:
    - Replace 'R' by 'Y'
    - Replace missing values by 'N'

*Note:* Both in subjects 3 and 4 there is an 'F' erroneous value assigned to one sample in this attribute, so we replaced it by 'N'.

6- [LiWeCool] Homogenize all qualification's attributes.

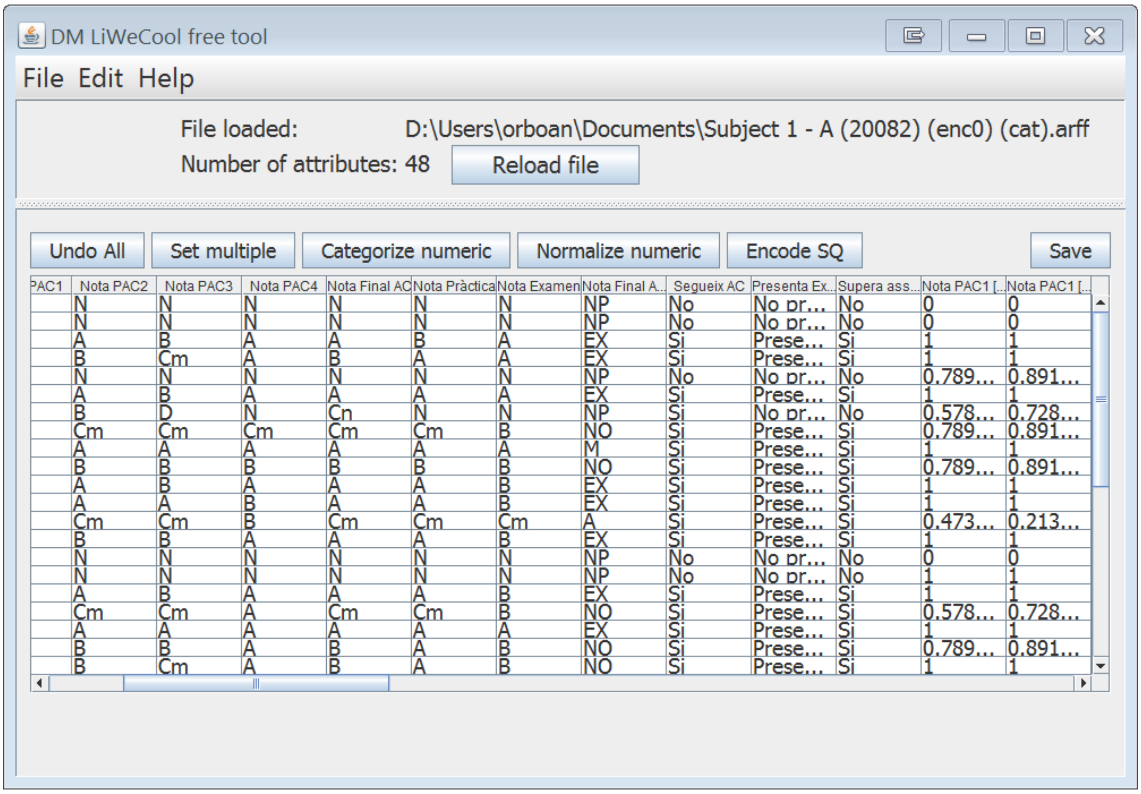
- Subjects 3 and 4: some are fully numeric but others has mixed values (numerical and nominal). Specifically, some have the 'NP' value for not delivered CE works, practicums or other activities, instead of a 0 as there is in the fully numerical student qualifications' attributes. So, replace 'NP' (nominal) by 0 (zero number). All SQ attributes are now fully numeric.

7- [LiWeCool] Preprocessing data.

Comments:

- the *klog* normalization for 'Nota Examen' (in subjects 3 and 4) were only experimented with non-'CE successful' instances ('Nota Final AC' = SU) because it is the only status in which a 'Nota Examen' lower than 2 leads to a non-success final subject qualification (while if it is greater than 2 then the subject successfulness depends on practicum qualifications). For students with the successful 'Nota Final AC' = AP (succesfully passed the CE) it is possible to gain a success final subject qualification without taking the exam, thus we have no prior domain information regarding any threshold value. Only for students who have passed the CE but their qualifications are lower than 6.5 there is an obligation for taking the exam, but in that case the successful exam threshold is variable.

– As the provided nominal qualifications in subjects 1 and 2 will be *klog-encoded*, the numeric qualifications in subjects 3 and 4 should be *klog-normalized* to put all subjects at the same starting conditions and, therefore, the results in the SUB (subject) dimension will provide comparable and not biased results.



File loaded: D:\Users\orboan\Documents\Subject 1 - A (20082) (enc0) (cat).arff

Number of attributes: 48

| PAC1 | Nota PAC2 | Nota PAC3 | Nota PAC4 | Nota Final AC | Nota Pràctica | Nota Examen | Nota Final A... | Segueix AC | Presenta Ex... | Supera ass... | Nota PAC1 [... | Nota PAC1 [... |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | N | N | N | N | N | N | NP | No | No pr... | No | 0 | 0 |
|  | N | N | N | N | N | N | NP | No | No pr... | No | 0 | 0 |
|  | A | B | A | A | B | A | EX | Si | Prese... | Si | 1 | 1 |
|  | B | Cm | A | B | A | A | EX | Si | Prese... | Si | 1 | 1 |
|  | N | N | N | N | N | N | NP | No | No pr... | No | 0.789... | 0.891... |
|  | A | B | A | A | A | A | EX | Si | Prese... | Si | 1 | 1 |
|  | B | D | N | Cn | N | N | NP | Si | No pr... | No | 0.578... | 0.728... |
|  | Cm | Cm | Cm | Cm | Cm | B | NO | Si | Prese... | Si | 0.789... | 0.891... |
|  | A | A | A | A | A | A | M | Si | Prese... | Si | 1 | 1 |
|  | B | B | B | B | B | B | NO | Si | Prese... | Si | 0.789... | 0.891... |
|  | A | B | A | A | A | B | EX | Si | Prese... | Si | 1 | 1 |
|  | A | A | B | A | A | B | EX | Si | Prese... | Si | 1 | 1 |
|  | Cm | Cm | B | Cm | Cm | Cm | A | Si | Prese... | Si | 0.473... | 0.213... |
|  | B | B | A | A | A | B | EX | Si | Prese... | Si | 1 | 1 |
|  | N | N | N | N | N | N | NP | No | No pr... | No | 0 | 0 |
|  | N | N | N | N | N | N | NP | No | No pr... | No | 1 | 1 |
|  | A | B | A | A | A | B | EX | Si | Prese... | Si | 1 | 1 |
|  | Cm | Cm | A | Cm | Cm | B | NO | Si | Prese... | Si | 0.578... | 0.728... |
|  | A | A | A | A | A | A | EX | Si | Prese... | Si | 1 | 1 |
|  | B | B | A | B | A | B | NO | Si | Prese... | Si | 0.789... | 0.891... |
|  | B | Cm | A | B | A | B | NO | Si | Prese... | Si | 1 | 1 |

*LiWeCool sample screenshot*

8- [Weka] Additional tasks:

– For subjects 3 and 4: Extracting all student instances with a (un)successful final CE qualification. The goal here is to divide all instances in two subsets, because the final subject qualification is achieved in different manners depending on whether the CE is passed or not (with a threshold value of 3.5). To do this task the LiWeCool-generated ARFF file (with all instances), after cleansing and preprocessing, has to be loaded in Weka, where we make use of the unsupervised and instance filter named SubSetByExpression.

9- [Weka] Ponderation of numeric qualifications for students who has successfully passed the CE (subjects 3 and 4). This ponderation follows the next domain rules:

> 1. Nota Final AC = 0,35*Nota PAC1 + 0,35*Nota PAC2 + 0,30*Nota Activitat
>
> 2. Nota Final Assignatura
>
> IF (Nota Final AC >= 3,5) THEN Nota Final Assignatura = 0,6*(Nota Examen + Nota Pract1 + Nota Pract2) + 0,4*(Nota Final AC)
>
> IF (Nota Final AC < 3,5) THEN Nota Final Assignatura = Nota Examen + Nota Pract1 + Nota Pract2

To do this ponderation, the `unsupervised.attribute.AddExpression` filter were used.

Therefore, an additional goal to those already established in section *2.4-Summary of the goals*, is to test the differences, if any, between weighted and not weighted data.

10- [LiWeCool] The final preprocessing task were done using LiWeCool again. All subjects should have the following dichotomized attributes to make possible to test the quality of clustering results:

– The first has to answer the question: has the student successfully passed the subject or not? For subjects 1 and 2 this attribute is already provided (named 'Supera Assignatura') but in subjects 3 and 4 we have a numeric attribute with the subject's final qualifications. In this case, it was necessary to categorize this numeric attribute to create the new dichotomized one, with the threshold at 5.0 (the numeric qualifications are given in a [0,10] scale with successfully passing the subject from 5.0 to upwards).

– The second attribute has to answer whether the student has abandoned the subject or not. In subjects 1 and 2 (the UOC subjects) there is a nominal final subject qualification with possible values {M, EX, NO, AP, SU, NP} where NP represents the student has left the subject, thus the creation of this new dichotomic attribute can be achieved by means of replacing every nominal non equal to NP with the same value. In the other hand, the subjects 3 and 4 provides the mentioned numeric attribute for final subject qualifications, but this attribute was in fact and originally a mixture of numeric and nominals, where only the students who abandoned the subject are marked with the NP nominal value; thus, all numeric values has to be replaced by a unique nominal not equal to NP (note here that the NP nominal value were replaced by a zero numeric value in a previous step, but this is meaningless for this dichotomization).

### 4.3- Measuring the quality of results

These last two dichotomic attributes are called *target attributes* because the clustering results measure their quality with the ratio of successful matching instances from the total clustered instances. To be more precise, this means:

– If applied clustering predicts instance X as to be a successful student (X is classified in the cluster of successful students) then:

- if the target attribute says X is a successful student, then the matching rate increases by 1/N (where N is the total number of instances processed by the clustering algorithm), thus the quality of the key attributes as predictors increases;

- if the target attribute says X in NOT a successful student, then the matching rate does not increase, thus the quality of the key attributes as predictors do not increase.

Weka includes the option to automatically calculate this matching rate (similarly as how a discriminant analysis does it), indicating one target nominal attribute (in fact, it calculates the non-matching rate as a percentage, from which we automatically obtain the matching rate with a simple math operation when introducing the results in a spreadsheet).

### 4.4- Ready-to-Weka files

At the end of the data pre-processing phase we obtained its output, the ready-to-Weka ARFF files, which is in turn the input for the next phase 'Clustering the data'. These files are:

| Subject | Ready-to-Weka files (preprocessed data) |
|---|---|
| 1 | Subject 1 - A (20081) (enc0) (cat).arff<br>Subject 1 - A (20082) (enc0) (cat).arff |
| 2 | Subject 2 - A (20081) (enc0) (cat).arff<br>Subject 2 - A (20082) (enc0) (cat).arff |
| 3 | Subject 3 (enc) (cat) (full).arff<br>Subject 3 (enc) (cat) (sce).arff<br>Subject 3 (enc) (cat) (sce) (pon).xrff<br>Subject 3 (enc) (cat) (uce).arff |
| 4 | Subject 4 (enc) (cat) (full).arff |

*Filename description:*

*Subject X (200XN) – [A] [(pond)] [(enc[0|1])] [(cat)] [(full|sce|uce)].{arff|xrff}*

Where:

Subject X: We have 4 subjects in our case of study, tagged with numbers 1 to 4. *'X'* stands for the subject number.

(200XN): The academic semester, where 200X is the year, and N = {1|2} indicates whether it's the first or the second semester. For the non-UOC subjects we have no such information.

A: This means the file contains only the basic data set, without extended data.

(pond): The 'PAC' (CE works) attributes from non-UOC subjects are weighted with the rules mentioned above.

(enc[0|1]):

- (enc0): Nominal qualifications are encoded with some factor k values.
- (enc1): Numeric qualifications are normalized with some factor k values.
- (enc): Both enc0 and enc1.

(cat): when file contents categorized or dicothomized attributes.

(full|sce|uce): For non-UOC subjects:

- full: includes all samples
- sce : includes samples with successful continous evaluation final qualification ('Nota Final AC'>=3.5)
- uce: full – sce

## 4.5- Goals and domain hypotheses

Before proceeding with the clustering of data, at this point a summary of the domain hypotheses or domain stated problems and the other goals we are seeking for, in this case of study, is advisable:

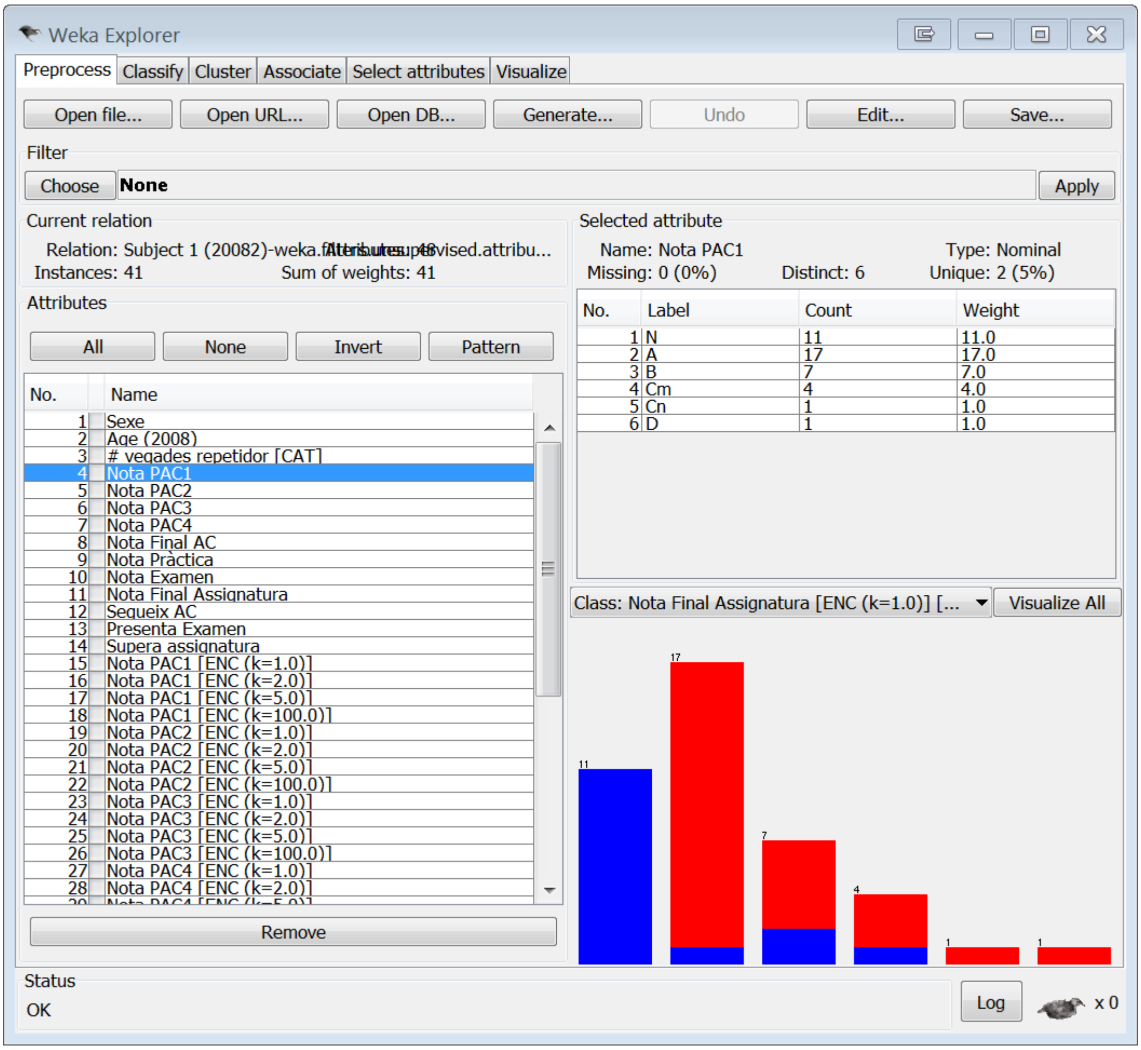
| *Id* | *Goal or hypothesis / problem to test* |
|---|---|
| 1 | To find out what combination of key attributes best work as predictors for:<br>– the student's successfulness;<br>– and the student's abandonment rate. |
| 2 | To compare the three selected clustering algorithms and determine which best fit to our domain-related problems:<br>– Simple K-means<br>– Hierarchical clustering<br>– Expectation-Maximization |
| 3 | To test how the klog function affects the quality of results. |
| 4 | To test the differences in the results, if any, between weighted and not weighted data. |
| 5 | To compare subjects' results and determine whether the quality of predictors are similar or not, even if different combination of key attributes are best recommended as predictors when changing from one subject to another. |

## 4.6- Clustering the data

*Clustering*

The clustering processes have already been described, but we show now how are carried out. The steps are as follows, taking as an example the subject 1 (academic semester 20082):

1- Loading the *Subject 1 - A (20082) (enc0) (cat).arff* file into Weka.



Filter: None

Current relation
Relation: Subject 1 (20082)-weka.filters.unsupervised.attribu... Attributes: 48
Instances: 41 Sum of weights: 41

Selected attribute
Name: Nota PAC1 Type: Nominal
Missing: 0 (0%) Distinct: 6 Unique: 2 (5%)

| No. | Label | Count | Weight |
|---|---|---|---|
| 1 | N | 11 | 11.0 |
| 2 | A | 17 | 17.0 |
| 3 | B | 7 | 7.0 |
| 4 | Cm | 4 | 4.0 |
| 5 | Cn | 1 | 1.0 |
| 6 | D | 1 | 1.0 |

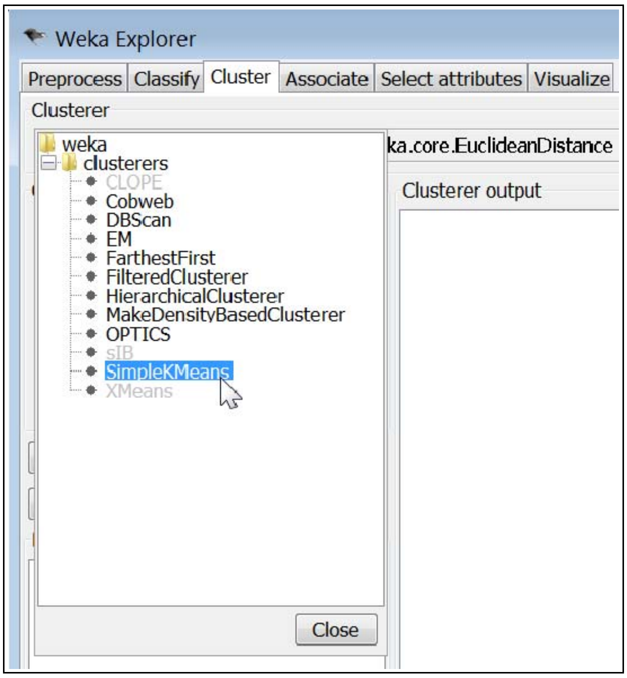
Attributes

| No. | Name |
|---|---|
| 1 | Sexe |
| 2 | Age (2008) |
| 3 | # vegades repetidor [CAT] |
| 4 | Nota PAC1 |
| 5 | Nota PAC2 |
| 6 | Nota PAC3 |
| 7 | Nota PAC4 |
| 8 | Nota Final AC |
| 9 | Nota Pràctica |
| 10 | Nota Examen |
| 11 | Nota Final Assignatura |
| 12 | Segueix AC |
| 13 | Presenta Examen |
| 14 | Supera assignatura |
| 15 | Nota PAC1 [ENC (k=1.0)] |
| 16 | Nota PAC1 [ENC (k=2.0)] |
| 17 | Nota PAC1 [ENC (k=5.0)] |
| 18 | Nota PAC1 [ENC (k=100.0)] |
| 19 | Nota PAC2 [ENC (k=1.0)] |
| 20 | Nota PAC2 [ENC (k=2.0)] |
| 21 | Nota PAC2 [ENC (k=5.0)] |
| 22 | Nota PAC2 [ENC (k=100.0)] |
| 23 | Nota PAC3 [ENC (k=1.0)] |
| 24 | Nota PAC3 [ENC (k=2.0)] |
| 25 | Nota PAC3 [ENC (k=5.0)] |
| 26 | Nota PAC3 [ENC (k=100.0)] |
| 27 | Nota PAC4 [ENC (k=1.0)] |
| 28 | Nota PAC4 [ENC (k=2.0)] |

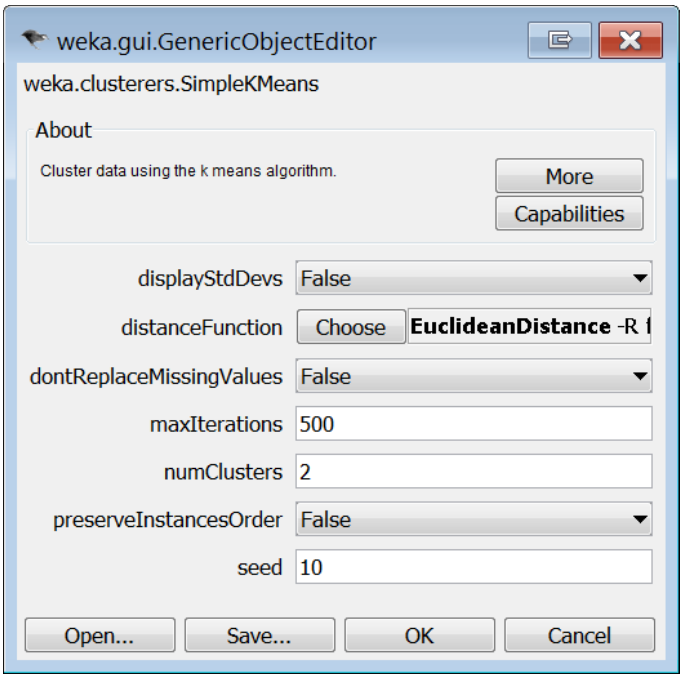
Class: Nota Final Assignatura [ENC (k=1.0)] [...

Status
OK

2- Selecting the chosen clustering algorithm:



3- Configuring the parameters for running properly the selected clustering algorithm:



weka.clusterers.SimpleKMeans

Cluster data using the k means algorithm.

displayStdDevs False

distanceFunction EuclideanDistance -R f

dontReplaceMissingValues False

maxIterations 500

numClusters 2

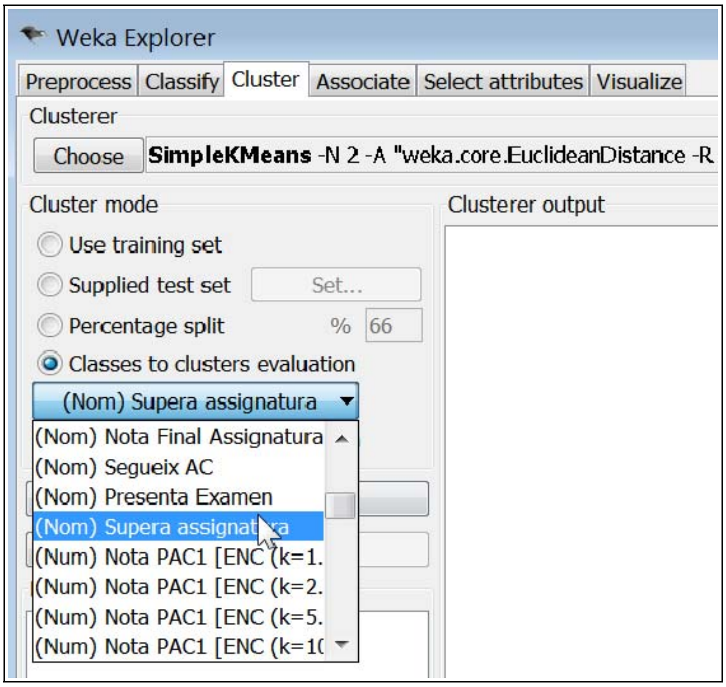
preserveInstancesOrder False

seed 10

4- Selecting the target attribute:

– 'Supera assignatura' when looking for students' successfulness predictors.

– 'Nota Final Assignatura [CAT2]' ("categorized 2" is the used code by the researchers when a numerical qualifications attribute is dichotomized into {NP, P} (abandoned the subject, not abandoned).



SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R

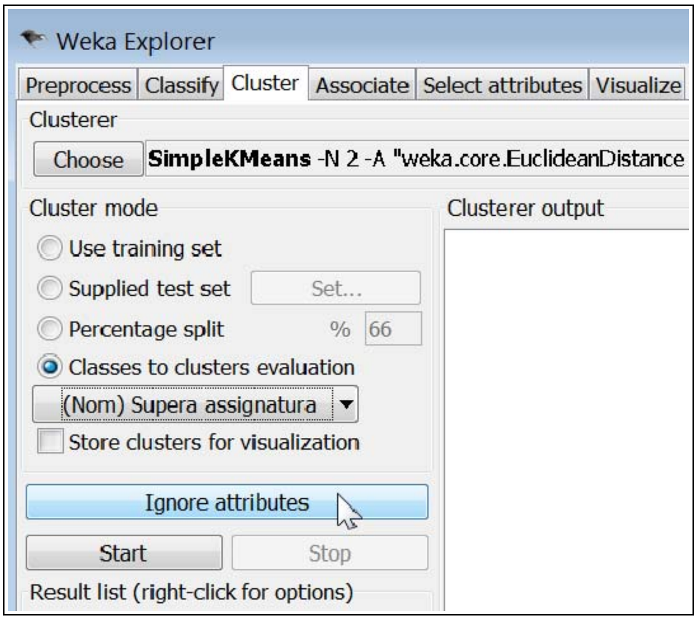
● Classes to clusters evaluation

(Nom) Supera assignatura

(Nom) Nota Final Assignatura
(Nom) Segueix AC
(Nom) Presenta Examen
(Nom) Supera assignatura
(Num) Nota PAC1 [ENC (k=1.
(Num) Nota PAC1 [ENC (k=2.
(Num) Nota PAC1 [ENC (k=5.
(Num) Nota PAC1 [ENC (k=1(

5- Selecting the key attributes:

By default, Weka includes all attributes loaded with the file, thus if a combination of key attributes has to be selected, we must tell to Weka to ignore all other attributes:



Weka Explorer

Clusterer

SimpleKMeans -N 2 -A "weka.core.EuclideanDistance

Cluster mode

○ Use training set

○ Supplied test set

○ Percentage split % 66

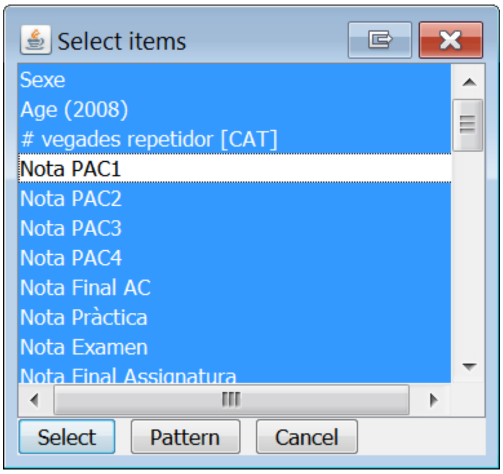
● Classes to clusters evaluation

(Nom) Supera assignatura

□ Store clusters for visualization

Clusterer output

Result list (right-click for options)



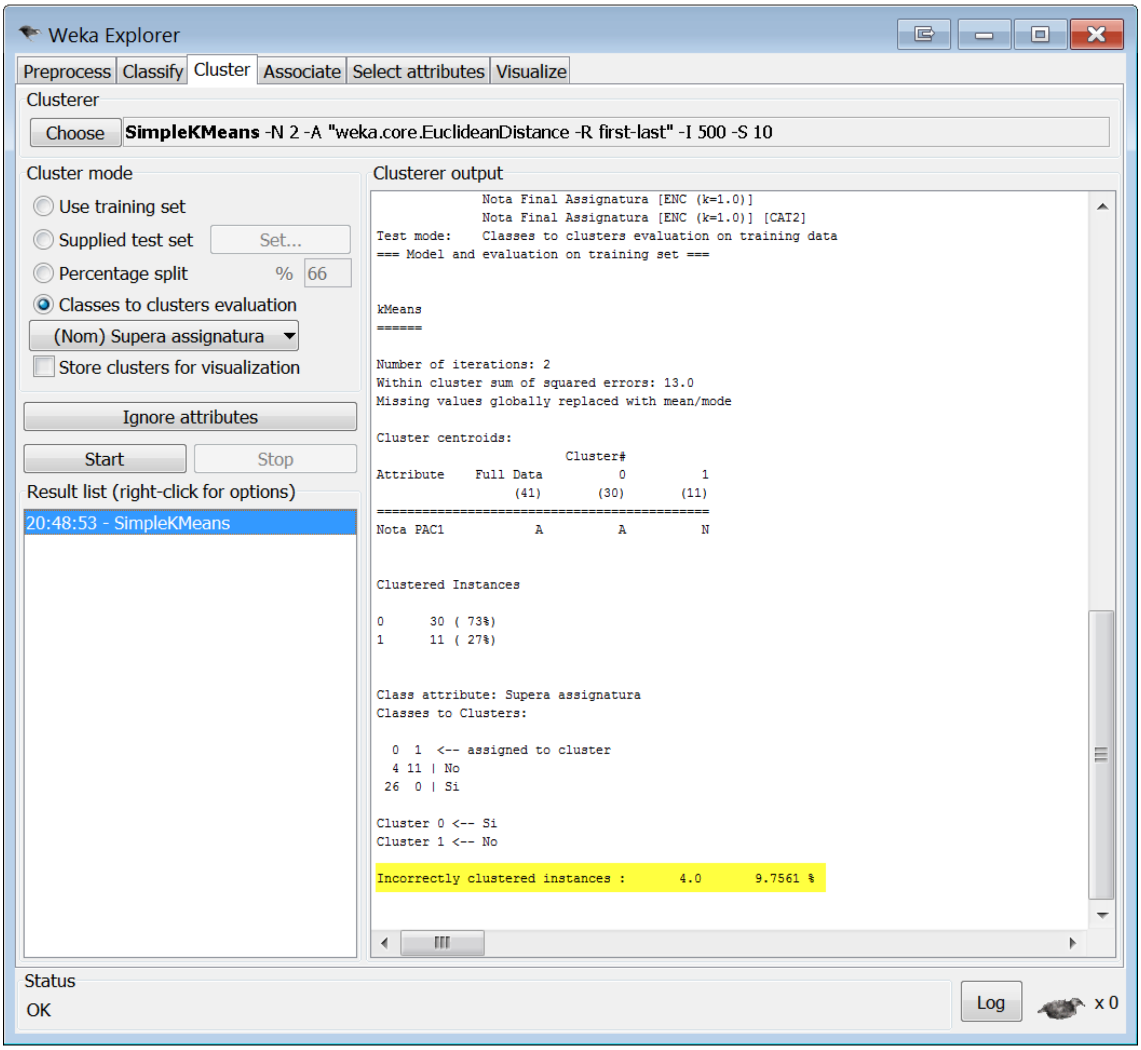
Select items

Sexe
Age (2008)
# vegades repetidor [CAT]
Nota PAC1
Nota PAC2
Nota PAC3
Nota PAC4
Nota Final AC
Nota Pràctica
Nota Examen
Nota Final Assignatura

As it is shown, all attributes are ignored but the 'Nota PAC1', this means 'Nota PAC1' is the key attribute.

6- Apply and run the clustering algorithm

Clicking the 'Start' button the algorithm runs. The clustered output results are offered:



Clusterer: SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10

Cluster mode:

○ Use training set

○ Supplied test set

○ Percentage split % 66

● Classes to clusters evaluation

(Nom) Supera assignatura

□ Store clusters for visualization

Result list: 20:48:53 - SimpleKMeans

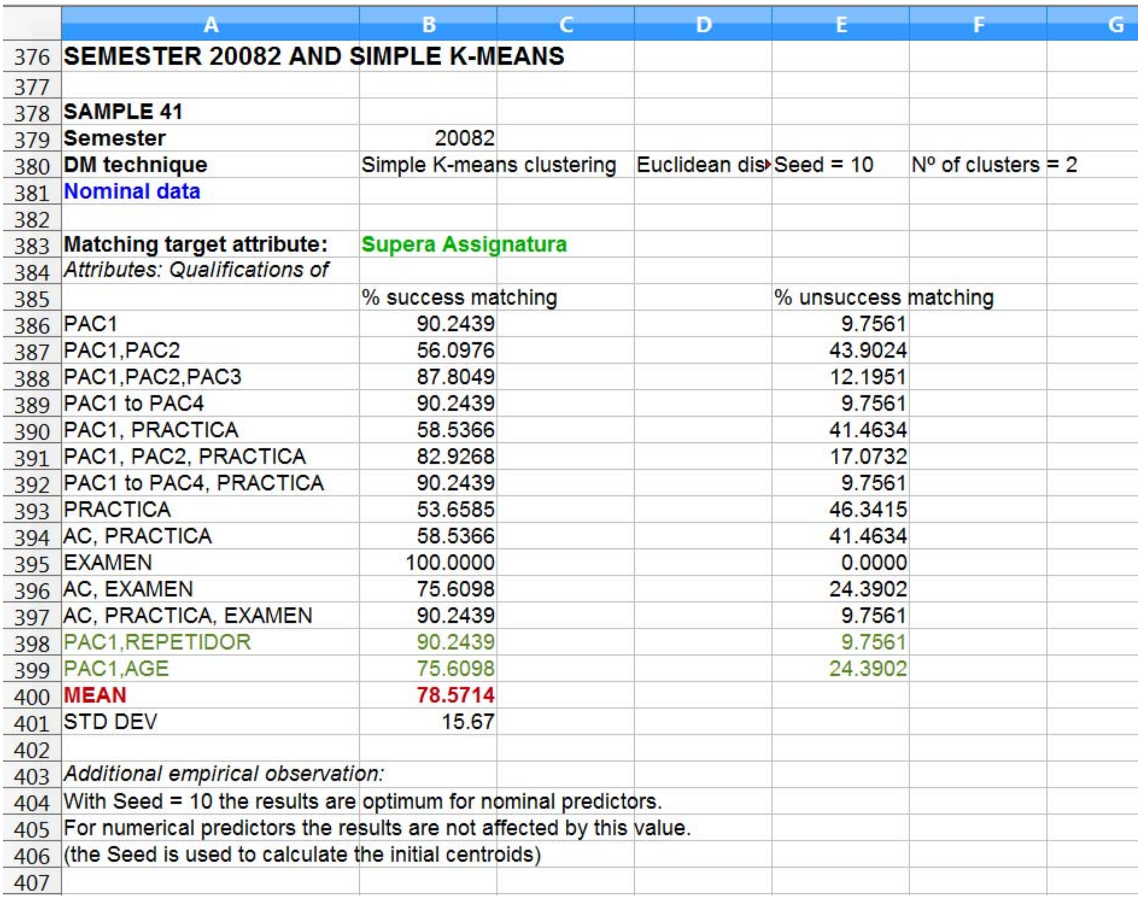
```
              Nota Final Assignatura [ENC (k=1.0)]
              Nota Final Assignatura [ENC (k=1.0)] [CAT2]
Test mode:    Classes to clusters evaluation on training data
=== Model and evaluation on training set ===


kMeans
======

Number of iterations: 2
Within cluster sum of squared errors: 13.0
Missing values globally replaced with mean/mode

Cluster centroids:
                        Cluster#
Attribute    Full Data         0          1
                  (41)       (30)       (11)
============================================
Nota PAC1            A          A          N



Clustered Instances

0      30 ( 73%)
1      11 ( 27%)


Class attribute: Supera assignatura
Classes to Clusters:

  0  1  <-- assigned to cluster
  4 11 | No
 26  0 | Si

Cluster 0 <-- Si
Cluster 1 <-- No

Incorrectly clustered instances :      4.0       9.7561 %
```

Status: OK

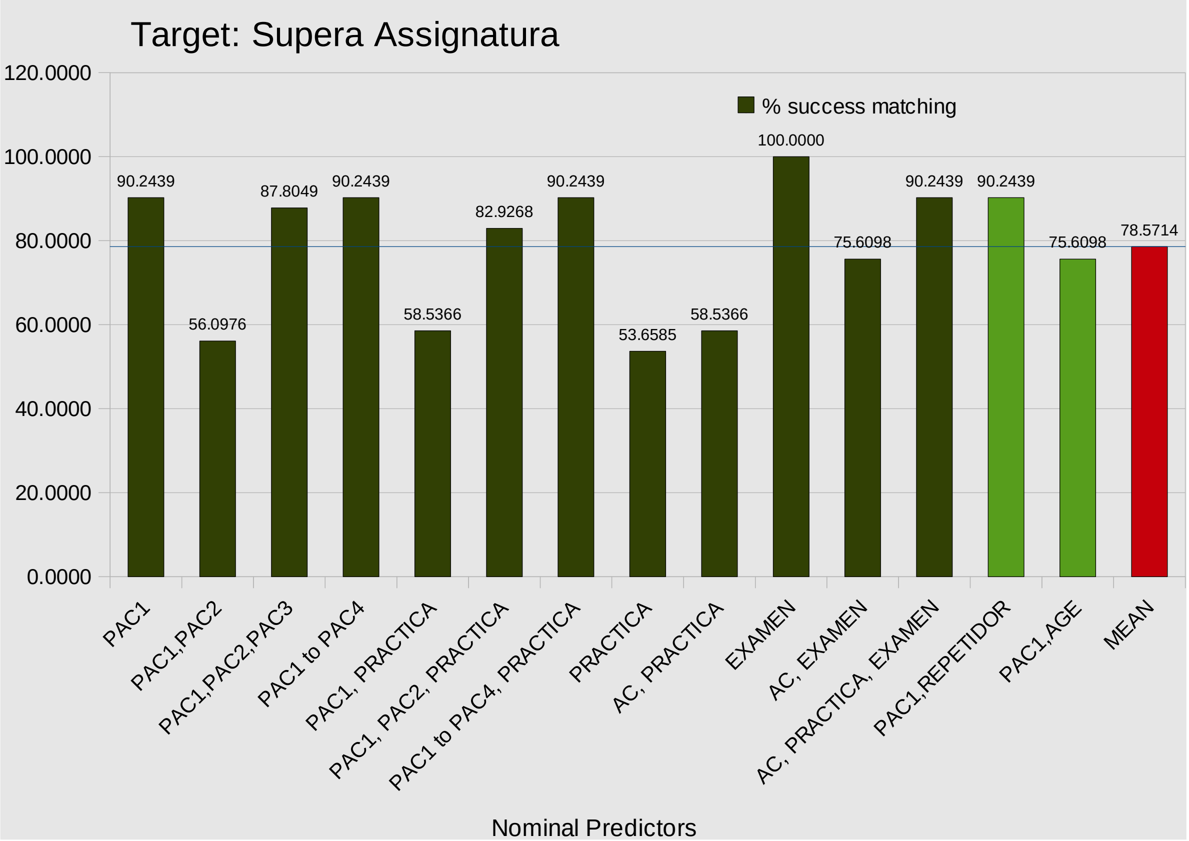
7- Entering the non-matching rate into the results spreadsheet file, as it is shown later.

8- Repeat steps 5 to 7 for every combination of key attributes wanted to be tested as a possible target predictors.

At this point the following sample of results is finished in the results spreadsheet file:



| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 376 | **SEMESTER 20082 AND SIMPLE K-MEANS** | | | | | | |
| 377 | | | | | | | |
| 378 | **SAMPLE 41** | | | | | | |
| 379 | **Semester** | 20082 | | | | | |
| 380 | **DM technique** | Simple K-means clustering | | Euclidean dis | Seed = 10 | Nº of clusters = 2 | |
| 381 | **Nominal data** | | | | | | |
| 382 | | | | | | | |
| 383 | **Matching target attribute:** | **Supera Assignatura** | | | | | |
| 384 | *Attributes: Qualifications of* | | | | | | |
| 385 | | % success matching | | | % unsuccess matching | | |
| 386 | PAC1 | 90.2439 | | | 9.7561 | | |
| 387 | PAC1,PAC2 | 56.0976 | | | 43.9024 | | |
| 388 | PAC1,PAC2,PAC3 | 87.8049 | | | 12.1951 | | |
| 389 | PAC1 to PAC4 | 90.2439 | | | 9.7561 | | |
| 390 | PAC1, PRACTICA | 58.5366 | | | 41.4634 | | |
| 391 | PAC1, PAC2, PRACTICA | 82.9268 | | | 17.0732 | | |
| 392 | PAC1 to PAC4, PRACTICA | 90.2439 | | | 9.7561 | | |
| 393 | PRACTICA | 53.6585 | | | 46.3415 | | |
| 394 | AC, PRACTICA | 58.5366 | | | 41.4634 | | |
| 395 | EXAMEN | 100.0000 | | | 0.0000 | | |
| 396 | AC, EXAMEN | 75.6098 | | | 24.3902 | | |
| 397 | AC, PRACTICA, EXAMEN | 90.2439 | | | 9.7561 | | |
| 398 | PAC1,REPETIDOR | 90.2439 | | | 9.7561 | | |
| 399 | PAC1,AGE | 75.6098 | | | 24.3902 | | |
| 400 | **MEAN** | **78.5714** | | | | | |
| 401 | STD DEV | 15.67 | | | | | |
| 402 | | | | | | | |
| 403 | *Additional empirical observation:* | | | | | | |
| 404 | With Seed = 10 the results are optimum for nominal predictors. | | | | | | |
| 405 | For numerical predictors the results are not affected by this value. | | | | | | |
| 406 | (the Seed is used to calculate the initial centroids) | | | | | | |
| 407 | | | | | | | |



9- Repeat steps 2 to 8 selecting the same key attributes but klog processed or weighted (in the cases that these transformations apply).

10- Repeat steps 2 to 9 for every the three different clustering algorithms and for every different values for their configuration parameters. The hierarchical agglomerative algorithm were repeated once for the Single Link and once for the Mean-Link.

11- Repeat steps 1 to 10 for every *Ready-to-Weka* file obtained from the previous data pre-processing phase.

### 4.5- Quantitative results

Next a summarized version of results is shown. The complete set of results, with visualization histograms, can be found at the **Results.ods** file (contact author for requesting it). The key attributes selected to be included here are only a little part of all considered combinations, and are the ones with highest rates while corresponding to early CE stages when the heterogeneous student groups have still time to be created / redistributed in the current semester. The Mean value belongs to all considered key attributes (found in the *Results.ods* file) not only to those that are shown here. K-means and hierarchical agglomerative algorithms use the Euclidean distance.

Subject 1 [20081]
Target: Supera assignatura
Simple K-means (seed = 10)

| Key attributes | Matching rate (%) | | | | |
|---|---|---|---|---|---|
| | Nominal data | K = 1 | K = 2 | K = 5 | K = 100 |
| PAC1 | 79.07% | 81.39 | 83.72 | 83.72 | 83.72 |
| PAC1, PAC2 | 83.72 | 83.72 | 86.05 | 86.05 | 83.72 |
| PAC1, PRACTICA | 93.02 | 93.02 | 88.37 | 86.05 | 86.05 |
| PAC1, PAC2, PRACTICA | 90.7 | 93.02 | 90.7 | 90.7 | 90.7 |
| PRACTICA | 74.41 | 93.02 | 93.02 | 93.02 | 93.02 |
| MEAN | 78.74 | 87.87 | 88.2 | 87.54 | 87.38 |

Subject 1 [20081]
Target: Abandona assignatura
Simple K-means (seed = 10)

| Key attributes | Matching rate (%) | | | | |
|---|---|---|---|---|---|
| | Nominal data | K = 1 | K = 2 | K = 5 | K = 100 |
| PAC1 | 81.40% | 83.72 | 81.4 | 81.4 | 81.4 |
| PAC1, PAC2 | 81.4 | 81.4 | 83.72 | 83.72 | 81.4 |
| PAC1, PRACTICA | 95.35 | 95.34 | 86.05 | 83.72 | 83.72 |
| PAC1, PAC2, PRACTICA | 90.7 | 86.04 | 88.37 | 88.37 | 88.37 |
| PRACTICA | 67.44 | 90.7 | 90.7 | 90.7 | 90.7 |
| MEAN | 80 | 86.66 | 85.12 | 84.5 | 84.34 |

Subject 1 [20081]
Target: Supera assignatura
Hierarchical clustering (Simple Linkage)

| Key attributes | Matching rate (%) | | |
|---|---|---|---|
| | Nominal data | K = 1 | K = 100 |
| PAC1 | 58.14% | 79.07 | 83.72 |
| PAC1, PAC2 | 76.74 | 58.14 | 83.72 |
| PAC1, PRACTICA | 79.06 | 55.81 | 60.47 |
| PAC1, PAC2, PRACTICA | 74.42 | 58.14 | 86.05 |
| PRACTICA | 93.02 | 93.02 | 93.02 |
| MEAN | 74.42 | 74.25 | 82.39 |

Subject 1 [20081]
Target: Abandona assignatura
Hierarchical clustering (Simple Linkage)

| Key attributes | Matching rate (%) | | |
|---|---|---|---|
| | Nominal data | K = 1 | K = 100 |
| PAC1 | 65.11% | 81.39% | 81.39% |
| PAC1, PAC2 | 79.07 | 65.11 | 81.39% |
| PAC1, PRACTICA | 86.05 | 58.14 | 67.44 |
| PAC1, PAC2, PRACTICA | 67.44 | 65.12 | 83.72 |
| PRACTICA | 67.44 | 100 | 90.7 |
| MEAN | 79.53 | 82.17 | 82.17 |

Subject 1 [20081]
Target: Supera assignatura
Expectation-Maximization (seed = 100)

| Key attributes | Matching rate (%) | | |
|---|---|---|---|
| | Nominal data | K = 1 | K = 100 |
| PAC1 | 55.81% | 81.4 | 83.72 |
| PAC1, PAC2 | 67.44 | 83.72 | 83.72 |
| PAC1, PRACTICA | 74.42 | 93.02 | 93.02 |
| PAC1, PAC2, PRACTICA | 74.42 | 83.72 | 90.7 |
| PRACTICA | 55.81 | 93.02 | 93.02 |
| MEAN | 69.27 | 85.55 | 87.87 |

Subject 1 [20081]
Target: Abandona assignatura
Expectation-Maximization (seed = 100)

| Key attributes | Matching rate (%) | | |
|---|---|---|---|
| | Nominal data | K = 1 | K = 100 |
| PAC1 | 62.79% | 83.72 | 81.39 |
| PAC1, PAC2 | 60.47 | 81.4 | 76.74 |
| PAC1, PRACTICA | 67.44 | 95.35 | 90.7 |
| PAC1, PAC2, PRACTICA | 67.44 | 76.74 | 88.37 |
| PRACTICA | 62.79 | 100 | 90.7 |
| MEAN | 72.71 | 86.51 | 84.5 |

Subject 2 [20081]
Target: Supera assignatura
Simple K-means (seed = 10)

| Key attributes | Matching rate (%) | | |
|---|---|---|---|
| | Nominal data | K = 1 | K = 100 |
| PAC1 | 64.37% | 68.97 | 68.97 |
| PAC1, PAC2 | 55.17 | 86.21 | 80.46 |
| PAC1, PAC2, PAC3 | 50.57 | 85.06 | 80.46 |
| MEAN | 68.39 | 80 | 79.89 |

Subject 2 [20081]
Target: Abandona assignatura
Simple K-means (seed = 10)

| Key attributes | Matching rate (%) | | |
|---|---|---|---|
| | Nominal data | K = 1 | K = 100 |
| PAC1 | 74.71 | 77.01 | 77.01 |
| PAC1, PAC2 | 55.17 | 85.06 | 79.31 |
| PAC1, PAC2, PAC3 | 59.77 | 88.51 | 79.31 |
| MEAN | 67.77 | 81.9 | 77.77 |

Subject 2 [20081]
Target: Supera assignatura
Hierarchical clustering (Simple Linkage)

| Key attributes | Matching rate (%) | | |
|---|---|---|---|
| | Nominal data | K = 1 | K = 100 |
| PAC1 | 60.92% | 64.37 | 68.97 |
| PAC1, PAC2 | 55.17 | 67.82 | 67.82 |
| PAC1, PAC2, PAC3 | 50.57 | 85.06 | 85.06 |
| MEAN | 66.32 | 68.97 | 72.76 |

Subject 2 [20081]
Target: Abandona assignatura
Hierarchical clustering (Simple Linkage)

| Key attributes | Matching rate (%) | | |
|---|---|---|---|
| | Nominal data | K = 1 | K = 100 |
| PAC1 | 62.07 | 74.71 | 77.01 |
| PAC1, PAC2 | 65.52 | 75.86 | 75.86 |
| PAC1, PAC2, PAC3 | 57.47 | 88.51 | 88.51 |
| MEAN | 74.48 | 76.67 | 78.16 |

Subject 2 [20081]
Target: Supera assignatura
Expectation-Maximization (seed = 100)

| Key attributes | Matching rate (%) | | |
|---|---|---|---|
| | Nominal data | K = 1 | K = 100 |
| PAC1 | 50.57% | 67.82 | 68.97 |
| PAC1, PAC2 | 85.06 | 87.36 | 80.46 |
| PAC1, PAC2, PAC3 | 85.06 | 83.91 | 83.91 |

| | | | |
|---|---|---|---|
| MEAN | 73.22 | 82.76 | 81.38 |

Subject 2 [20081]

Target: Abandona assignatura

Expectation-Maximization (seed = 100)

| Key attributes | Matching rate (%) | | |
|---|---|---|---|
| | Nominal data | K = 1 | K = 100 |
| PAC1 | 60.92 | 75.86 | 77.01 |
| PAC1, PAC2 | 86.21 | 86.21 | 79.31 |
| PAC1, PAC2, PAC3 | 88.51 | 87.36 | 87.36 |
| MEAN | 79.2 | 86.67 | 85.29 |

Subject 3 [unknown] [full]

Target: Supera assignatura

Simple K-means (seed = 10)

| Key attributes | Matching rate (%) | |
|---|---|---|
| | Original numeric data (k = 1) | K = 100 |
| PAC1 | 50.6 | 56.63 |
| PAC1, PAC2 | 59.04 | 56.63 |
| PAC1, PRACT1 | 55.42 | 56.63 |
| PAC1, PAC2, ACTIV | 56.62 | 63.86 |
| MEAN | 55.02 | 57.38 |

Subject 3 [unknown] [full]

Target: Abandona assignatura

Simple K-means (seed = 10)

| Key attributes | Matching rate (%) |
|---|---|
| | Original numeric data (k = 1) |
| PAC1 | 63.86 |
| PAC1, PAC2 | 60.24 |
| PAC1, PRACT1 | 66.27 |
| PAC1, PAC2, ACTIV | 60.24 |
| MEAN | 64.94 |

Subject 3 [unknown] [full]
Target: Supera assignatura
Hierarchical clustering (Simple Linkage)

| Key attributes | Matching rate (%) |
|---|---|
| | Original numeric data (k = 1) |
| PAC1 | 59.04 |
| PAC1, PAC2 | 77.11 |
| PAC1, PRACT1 | 77.11 |
| PAC1, PAC2, ACTIV | 77.11 |
| MEAN | 64.01 |

Subject 3 [unknown] [full]
Target: Abandona assignatura
Hierarchical clustering (Simple Linkage)

| Key attributes | Matching rate (%) |
|---|---|
| | Original numeric data (k = 1) |
| PAC1 | 73.49 |
| PAC1, PAC2 | 84.34 |
| PAC1, PRACT1 | 73.49 |
| PAC1, PAC2, ACTIV | 84.34 |
| MEAN | 73.37 |

Subject 3 [unknown] [full]
Target: Supera assignatura
Expectation-Maximization (seed = 100)

| Key attributes | Matching rate (%) |
|---|---|
| | Original numeric data (k = 1) |
| PAC1 | 55.42 |
| PAC1, PAC2 | 59.83 |
| PAC1, PRACT1 | 57.83 |
| PAC1, PAC2, ACTIV | 54.22 |
| MEAN | 55.29 |

Subject 3 [unknown] [full]
Target: Abandona assignatura
Expectation-Maximization (seed = 100)

| Key attributes | Matching rate (%) |
|---|---|
| | Original numeric data (k = 1) |
| PAC1 | 69.88 |
| PAC1, PAC2 | 57.83 |
| PAC1, PRACT1 | 72.29 |
| PAC1, PAC2, ACTIV | 60.24 |
| MEAN | 67.83 |

Subject 4 [unknown] [full]
Target: Supera assignatura
Simple K-means (seed = 10)

| Key attributes | Matching rate (%) | |
|---|---|---|
| | Original numeric data (k = 1) | K = 100 |
| PAC1 | 76.4 | 76.4 |
| PAC1, PAC2 | 75.28 | 77.53 |
| PAC1, PRACT1 | 65.17 | 67.42 |
| PAC1, PAC2, ACTIV | 73.03 | 75.28 |
| MEAN | 69.79 | 70.37 |

Subject 4 [unknown] [full]
Target: Abandona assignatura
Simple K-means (seed = 10)

| Key attributes | Matching rate (%) |
|---|---|
| | Original numeric data (k = 1) |
| PAC1 | 55.06 |
| PAC1, PAC2 | 51.69 |
| PAC1, PRACT1 | 82.02 |
| PAC1, PAC2, ACTIV | 53.93 |
| MEAN | 65.07 |

Subject 4 [unknown] [full]
Target: Supera assignatura
Hierarchical clustering (Simple Linkage)

| Key attributes | Matching rate (%) |
|---|---|
| | Original numeric data (k = 1) |
| PAC1 | 76.4 |
| PAC1, PAC2 | 77.11 |
| PAC1, PRACT1 | 74.16 |
| PAC1, PAC2, ACTIV | 61.8 |
| MEAN | 68.21 |

Subject 4 [unknown] [full]
Target: Abandona assignatura
Hierarchical clustering (Simple Linkage)

| Key attributes | Matching rate (%) |
|---|---|
| | Original numeric data (k = 1) |
| PAC1 | 57.3 |
| PAC1, PAC2 | 57.3 |
| PAC1, PRACT1 | 76.4 |
| PAC1, PAC2, ACTIV | 77.53 |
| MEAN | 64.15 |

Subject 4 [unknown] [full]
Target: Supera assignatura
Expectation-Maximization (seed = 100)

| Key attributes | Matching rate (%) |
|---|---|
| | Original numeric data (k = 1) |
| PAC1 | 76.4 |
| PAC1, PAC2 | 76.4 |
| PAC1, PRACT1 | 77.53 |
| PAC1, PAC2, ACTIV | 77.53 |
| MEAN | 74.03 |

Subject 4 [unknown] [full]
Target: Abandona assignatura
Expectation-Maximization (seed = 100)

| Key attributes | Matching rate (%) |
|---|---|
| | Original numeric data (k = 1) |
| PAC1 | 50.56 |
| PAC1, PAC2 | 57.3 |
| PAC1, PRACT1 | 53.93 |
| PAC1, PAC2, ACTIV | 53.93 |
| MEAN | 62.61 |

Additional empirical results obtained regarding the optimum values for the 'seed' parameters in the K-means and the EM algorithms: the used values of 10 (K-means) and 100 (EM) provides optimum results for both when the nominal attributes (for subjects 1 and 2) are processed, and for numerical key attributes it is tested that the results are not affected by this value.

Drawing the conclusions in the next section is done from the overall collection of results included in the Results.ods file, although it is already possible to get some clues from the summarized results shown in this section.

## 5- Conclusions

### 5.1- The predictors

*Target: Supera Assignatura*

In subjects 1 and 2, and from PACn key attributes, it is shown as a {PAC1, PAC2, PAC3} combination offers best predictive ratios, but only slightly upper than {PAC1, PAC2}. {PAC1} offers significant worst results instead. It is also stated that adding {PRACTICA} to any combination of the above, the matching ratios improve significantly in most samples (this does not apply to Subject 2 due it has no PRACTICA attribute).

For subjects 3 and 4 this can not be stated as clearly as with subjects 1 and 2, because the behaviour seems to not follow a definite pattern and without making use of any systematic statistical analysis, it appears as random (the addition of {PACn} with n > 1 or {PRACT1} or {ACTIV} enhances the ratio in some result samples, but not in others, and when they do, the differences are not very significant).

*Target: Abandona Assignatura*

For subject 1, any combination with the first three PAC attributes have very similar predictive ratios, and {PAC1, PAC2, PRACTICA} seems to improve significantly the only-PAC ratios. In subject 2, without the PRACTICA attribute, different behaviours are observed depending on the chosen algorithm, thus a global conclusion can not be achieve.

In subjects 3 and 4 a global conclusion can not be stated, too. The reason is the same: while for an algorithm the matching ratio increases as more later-in-time attributes are added, for another there is no difference, and for the other the ration decreases.

It has been also tested whether the inclusion of {Repetidor} as a key attribute has some effect on the results or not. The answer found is that this attribute at least does not improve the results (some are worst, some are unaltered with its addition). Therefore is difficult to infer any casualty relationship between the fact of being a repeater student and his or her successfulness or abandonment rate.

### 5.2- The algorithms

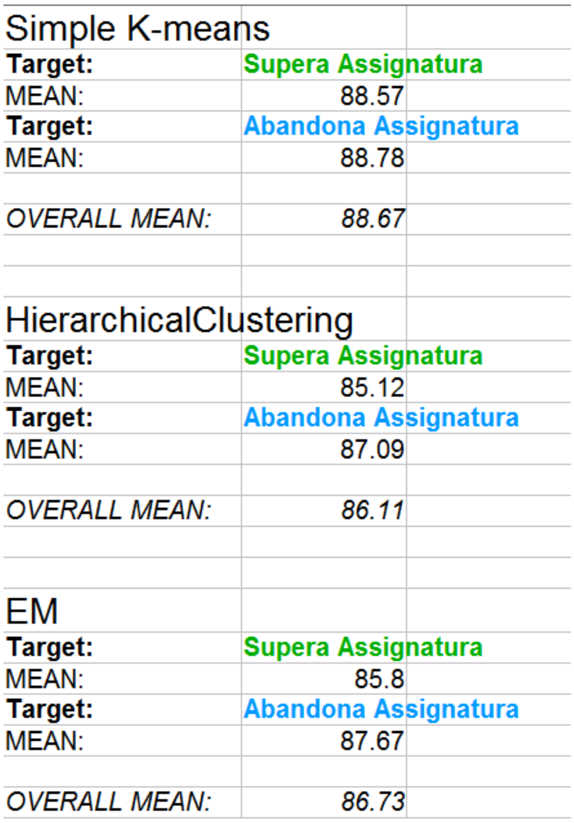
For the Subject 1, target 'Supera Assignatura' and for nominal data, while the best-to-worst ratio results order is K-means > Hierarchical > EM, for the academic semester 20081 it is Hierarchical > EM > K-means. Therefore, for nominal key attributes it is not possible to infer which is globally more effective.

When testing numeric qualifications, K-means, EM and Hierarchical with the Mean Linkage offer very similar results (are statistically equal) while Hierarchical with the Single Linkage the obtained ratios are clearly lower. The difference between using the Mean Linkage or the Single Linkage in the Hierarchical algorithm is not significant, though, when working with nominals.

The same conclusion can be extended to Subject 2 and to the 'Abandona assignatura' target.

When working with the non-UOC subjects, the only conclusion and curious is that here the Single Linkage gives us better results than the Mean Linkage. Concerning the comparison between algorithms, no sort can be done the better-to-worst order is very heterogeneous.

To illustrate what is here said, next the global success-ratio means for the subject 1 are shown:



| | |
|---|---|
| Simple K-means | |
| **Target:** | Supera Assignatura |
| MEAN: | 88.57 |
| **Target:** | Abandona Assignatura |
| MEAN: | 88.78 |
| | |
| *OVERALL MEAN:* | *88.67* |
| | |
| | |
| HierarchicalClustering | |
| **Target:** | Supera Assignatura |
| MEAN: | 85.12 |
| **Target:** | Abandona Assignatura |
| MEAN: | 87.09 |
| | |
| *OVERALL MEAN:* | *86.11* |
| | |
| | |
| EM | |
| **Target:** | Supera Assignatura |
| MEAN: | 85.8 |
| **Target:** | Abandona Assignatura |
| MEAN: | 87.67 |
| | |
| *OVERALL MEAN:* | *86.73* |

### 5.3- Numerical versus Nominal. The klog function.

When nominal qualifications (with no ordinal information) are encoded to numeric (domain ordinality injected) a main conclusion is reached here: the improvement for the success matching ratios are very significant (around a 10% or more). This happens either when targeting 'Supera Assignatura' and also 'Abandona Assignatura', thus it is very recommended to do such preprocessing on the data.

Injecting the additional successfulness information to the data through a *klog* transformation has show a slightly improvement for low values of k (2 and 5 were tested), but when k is bigger (100 were tested) than a still undetermined threshold, the results start to degrade. This evidence is logic in some way, because the numeric attribute tends to a dichotomized attribute while the k value increases, thus while the difference between successful and unsuccessful qualifications is emphasized (information increases) the difference between qualifications inside the same category (the full set is partitioned into two categories: successful and unsuccessful) decreases (so the self-contained information decreases). It has also been noted that *klog* transformation causes worst results when targeting 'Abandona Assignatura', and that could highly be due to the fact that the *klog* is constructed to emphasized successfulness but not the abandonment rate.

### 5.4- The subjects

On one side, Subject 1 offers the very best matching ratio results. Its ratio means are near the 90% of success in predicting the targets, and with some samples it reaches ratios very close to the full match (100%). The other UOC subject still give us good ratios but lower (around the 75% in mean).

On the other side, from the non-UOC subjects it is clearly more difficult to infer predictors because the ratios obtained are very worst (around the 60% in mean), being quite useless for predicting and inferring purposes unless more subjects (or academic semesters form the same subjects) are processed to achieve statistical confident results.

Additionally, we tested whether weighing the qualifications on a prior domain-related knowledge basis has any effect on the results or not. In this case there is no space for doubt: the ratios obtained with weighted data are equal than those obtained from non-weigthed data.

## 6- Future work

This research has only really started. There is plenty of possibilities to explore, and as for to be mentioned as examples, there are:

- To test more key attributes;
- To measure the time-consuming cost when running the algorithms (their time complexity);
- To test new parameters as different distances, different linkages;
- To test other clustering algorithms;
- To apply other DM techniques as association rules, classification;
- To experiment with alternative free Data mining tools.

Another pendent question is to find out the optimum value of k for the *klog* transformation, and how this optimum value of k depends on the DM technique applied and on the dataset. As it is almost impossible to proceed with this calculation manually, another future goal should be to extend the functionalities of the LiWeCool free tool to automatize this task.

# References

[1] The R-project for Statistical Computing, http://www.r-project.org/

[2] *Knowledge Discovery nuggets* (data mining comunity's top resource) http://www.kdnuggets.com/software/index.html

[3] Mehmed Kantardzic, *Data Mining: Concepts, Models, Methods, and Algorithms*, ed. Wiley-InterScience, IEEE Press, 2003.

[4] O.R. Zaïane, Recommender systems for e-learning: towards non-intrusive web mining, *Data Mining in e-learning*, Chapter 5. Ed. WIT Press (editors: C.Romero & S.Ventura), 2006.

[5] Willi Klosgen (editor), Jan M. Zytkow (editor), *Handbook of Data Mining and Knowledge Discovery*, ed. Oxford University Press Inc, 2002.

[6] P. Desikan, C. DeLong, K. Beemanapalli, A. Bose & J. Srivastava, Web Mining for self-directed e-learning, *Data Mining in e-learning*, Chapter 2. Ed. WIT Press (editors: C.Romero & S.Ventura), 2006.

[7] C. Pahl, Data Mining for the analysis of content interaction in web-based learning and training systems, *Data Mining in e-learning*, Chapter 3. Ed. WIT Press (editors: C.Romero & S.Ventura), 2006.

[8] F. Wang, On using data mining for browsing log analysis in learning environments, *Data Mining in e-learning*, Chapter 4. Ed. WIT Press (editors: C.Romero & S.Ventura), 2006.

[9] Joan Marc Carbó, Enric Mor, Julià Minguillón. *User navigational behaviour in e-learning virtual environments.* Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05), 2005. (Located at http://ppb.ined.uitm.edu.my/resources/journal/j4.pdf)

[10] Rosanna Costaguta. *Una revisión de Desarrollos Inteligentes para Aprendizaje Colaborativo soportado por Computadora*. Ed. Revista Ingeniería Informática, 2006. (Located at http://www.inf.udec.cl/~revista/ediciones/edicion13/articulo%2013-5.pdf).

[11] E. García Salcines, C. Romero Morales, S. Ventura Soto, C. De Castro Lozano. *Sistema recomendador colaborativo usando minería de datos distribuida para la mejora continua de cursos e-learning*. Ed. Revista IEEE-RITA, May 2008 (Located at http://webs.uvigo.es/cesei/RITA/200805/uploads/IEEE-RITA.2008.V3.N1.A3.pdf)

[12] Moodle de la Universidad de Zaragoza, http://moodle.unizar.es

[13] UOC virtual learning campus, http://www.uoc.edu

[14] ADL, *Sharable Content Object Reference Model* (SCORM) 2004, 2nd edn, overview, 2004.

[15] P. De Bra, Web-based Educational Hypermedia, *Data Mining in e-learning*, Chapter 1. Ed. WIT Press (editors: C.Romero & S.Ventura), 2006.

[16] Amelia Zafra, Sebastián ventura, *Predicting Student Grades in Learning Management Systems with Multiple Instance Genetic Programming*. EDM'09. 2nd International Conference on Eduational Data Mining, July 2009. (Located at http://www.educationaldatamining.org/EDM2009/uploads/proceedings/zafra)

[17] Cristóbal Romero, Sebastián Ventura, Enrique García, *Data Mining in course management systems: Moodle case study and tutorial*. ScienceDirect, Elsevier, 2007. (Located at http://sci2s.ugr.es/docencia/doctoM6/Romero-Ventura-Garcia-CE.pdf)

[18] Dietterich, T.G., Lathrop R.H., Lozano-Perez, T., *Solving the multiple instance problem with axis-parallel rectangles*, Artificial Intelligence, 89 (1-2), 31-71, 1997.

[19] Osmar R. Zaïane, *Building a Recommender Agent for e-learning systems* (Located at http://webdocs.cs.ualberta.ca/~zaiane/postscript/icce02.pdf)

[20] Jia Li, Osmar R. Zaïane. *Combining Usage, Content, and Structure Data to improve Web Site Recommendation*. (Located at http://webdocs.cs.ualberta.ca/~zaiane/postscript/ecweb04.pdf)

[21] Osmar R. Zaïane. *Web Usage Mining for a Better Web-Based Learning Environment*. (Located at http://webdocs.cs.ualberta.ca/TechReports/2001/TR01-05/TR01-05.pdf)

[22] Maurice D. Mulvenna, Sarabjot S. Anand, Alex G. Bücher. *Personalization on the Net using Web Mining*. 2000. (Located at http://www.mcbuchner.com/HTML/Research/PDF/CACM00.pdf)

[23] Enrique García, Cristóbal Romero, Carlos de Castro, Sebastián Ventura. *Usando minería de datos para la contínua mejora de cursos de e-learning*. Conferencia IADIS Ibero-Americana WWW/Internet, 2006. (Located at http://www.iadis.net/dl/final_uploads/200607L024.pdf).

[24] Enrique García, Cristóbal Romero, Carlos de Castro, Sebastián Ventura. *Sistema de Desarrollo Integrado para Cursos Hipermedia Adaptativos (INDESAHC)*. (Located at http://www.aipo.es/articulos/3/14.pdf)

[25] Cristóbal Romero, Sebastián Ventura, César Hervás. *Descubrimiento de Reglas de Predicción en Sistemas de e-learning utilizando Programación Genética*. 2005. (Located at http://www.lsi.us.es/redmidas/Capitulos/LMD05.pdf)

[26] Yueh-Min Huang, Juei-Nan Chen, Shu-Chen Cheng. *A method of Cross-level Frequent Pattern Mining for Web-based Instruction*. Journal of Educational Technology & Society Jul2007, Vol. 10 Issue 3, p305-319 - http://www.ifets.info/ , 2007. (Located at http://www.ifets.info/journals/10_3/21.pdf).

[27] Cristóbal Romero Morales, Sebastián Ventura Soto, Cesar Hervás Martínez. *Estado actual de la aplicación de la minería de datos a los sistemas de enseñanza basada en web*. CEDI 2005. III Taller de Minería de Satos y Aprendizaje, TAMIDA 2005. (Located at http://www.lsi.us.es/redmidas/CEDI/papers/189.pdf)

[28] C. Romero, S. Ventura. *Educational Data Mining: A survey from 1995 to 2005*. An Elsevier journal, 2006. (Located at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.702&rep=rep1&type=pdf)

[29] Maciej Kiewra. *Iterative Discovering of User's Preferences Using Web Mining*. International Journal of Computer Science & Applications, 2005. (Located at - http://www.tmrfindia.org/ijcsa/V2I25.pdf )

[30] H. Zhang, M. Spiliopoulou, B. Mobasher, C. Lee Giles, A. McCallum, O. Nasraoui, J. Srivastava, J. Yen (Eds.). *Advances in Web Mining and Web Usage Analysis*. 9th International Workshop on Knowledge Discovery on the Web, WebKDD 2007 and 1st International Workshop on Social Network Analysis, SNA-KDD 2007. San José, CA, USA. Revised papers. Ed. Springer, 2007.

[31] Winters, Titus deLaFayette, *Educational data mining: Collection of analysis of score matrices for outcomes-based assessment*. Dissertation. University of California, Riverside, 2006 (More info at http://proquest.umi.com/pqdweb?did=1192198831&Fmt=2&clientId=13807&RQT=309&VName=PQD)

[32] C. Romero Morales, S. Ventura Soto, C. De Castro Lozano. *Herramienta para el descubrimiento de Reglas de Predicción en Educación basada en web*. Congreso Intercción 2004 de AIPO, 2004. (Located at http://www.aipo.es/aipo/articulos/3/56.pdf)

[33] EATCO http://www.cpmti.tv/mediawiki/index.php?title=EATCO

[34] INDESAHC implementation http://www.cpmti.tv/mediawiki/index.php?title=INDESAHC

[35] CPMTI, http://www.cpmti.tv/mediawiki/index.php?title=CPMTI

[36] Red Española de Minería de Datos y Aprendizaje. http://www.lsi.us.es/redmidas/

[37] G. Holmes, A. Donkin, I.H. Witten. *Weka: A machine learning workbench*. 1994 (Located at http://www.cs.waikato.ac.nz/~ml/publications/1994/Holmes-ANZIIS-WEKA.pdf)

[38] TANAGRA, a free sotware data mining application, http://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html

[39] SALMS: SCORM-compliant Adaptive LMS, http://www.editlib.org/p/26933

[40] Different authors, *Data Mining in e-learning*, Ed. WIT Press (editors: C.Romero & S.Ventura), 2006.

[41] Michael Steinbach, George Karypis, Vipin Kumar. *A Comparison of Document Clustering Techniques*. KDD workshop on text mining, 2000. University of Minnesota. (Located at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.125.9225&rep=rep1&type=pdf).

[42] Sean Borman. *The Expectation Maximization Algorithm. A short tutorial*. Information Science Institute. University of Southern California. 2008. (Located at http://www.isi.edu/natural-language/teaching/cs562/2009/readings/B06.pdf).

[43] A.M. Moses, D.Y. Chiang, and M.B. Eisen. *Phylogenetic Motif Detection by Expectation-Maximization on Evolutionary Mixtures*. Pacific Symposium on Biocomputing 9:324-335 (2004). (Located at http://helix-web.stanford.edu/psb04/moses.pdf).

[44] David A. Wheeler. *Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the numbers!.* http://www.dwheeler.com/oss_fs_why.html . 2007.

[45] Wengang Liu. *Applying Educational Data Mining in E-learning Environment*. ARIES Lab Department of Computer Science. University of Saskatchewan.

[46] E Frank, M Hall, L Trigg, G Holmes, IH Witten. Data mining in bioinformatics using Weka. Bioinformatics, 2004. Oxford University Press. (Located at http://bioinformatics.oxfordjournals.org/cgi/reprint/20/15/2479.pdf)

[47] C. Da Cunha, B. Agard, A. Kusiak. *Data mining for improvement of product quality*. International Journal of Production Research, Volume 44. September 2006. (Located at http://www.informaworld.com/smpp/content~db=all~content=a749323840)

[48] Varun Kumar, Dharminder Kumar, R.K. Singh. *Outlier Mining in Medical Databases: An Application of Data Mining in Health Care Management to Detect Abnormal Values Presented In Medical Databases*. IJCSNS International Journal of Computer Science and Network Security. August 2008. (Located at http://paper.ijcsns.org/07_book/200808/20080838.pdf).

[49] Felix Jungermann. *Information Extraction with RapidMiner*. Proceedings of the GSCCL Symposium 'Sprachtechnologie und eHumanities' 2009. (Located at http://www-ai.informatik.uni-dortmund.de/DOKUMENTE/jungermann_2009a.pdf).

[50] The-Data-Mine site, http://www.the-data-mine.com/

[51] J. Alcalá-Fdez, L.Sánchez, S.García, M.J. Del Jesus, S.Ventura, J.M. Garrell, J.Otero, C.Romero, J.Bacardit, V.M.Rivas, J.C.Fernández, F.Herrera. *KEEL: a software tool to assess evolutionary algorithms for data-mining problems*. Springer-Verlag 2008. (Located at http://sci2s.ugr.es/keel/pdf/keel/articulo/Alcalaetal-SoftComputing-Keel1.0.pdf)

[52] Katharina Morik, Martin Scholz. *The MiningMart Approach to Knowledge Discovery in Databases*. Springer, 2004. (Located at http://www.springer.com/sgw/cda/pageitems/document/cda_downloaddocument/0,11855,0-0-45-124187-p18192908,00.pdf)

[53] MR Berthold, N Cebron, F Dill and others (book). *KNIME: The Konstanz information Miner*. Springer Berlin Heidelberg, 2008. (Located at http://www.springerlink.com/content/k2r4j835246276p7/)

[54] C. Sieb, T. Meinl, M. R. Berthold. *Parallel and Distributed data pipelining with Knime*. The Mediterranean Journal of Computers and Networks. SoftMotor Ltd. 2007. (Located at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.90.4256&rep=rep1&type=pdf)

[55] Rushing J, Ramachandran R, Nair U, Graves S, Welch R, Lin H (2005). *ADaM: a datamining toolkit for scientists and engineers.* Comput Geosci 31(5):607–618

[56] Ingo Mierswa, Ralf Klinkenberg, Simon Fischer, and Oliver Ritthoff. *A flexible Platform for Knowledge Discovery Experiments: YALE – Yet Another Learning Environment*. LLWA 03 – Tagungsband, 2003. (Located at http://www.kde.cs.uni-kassel.de/ws/LLWA03/fgml/final/poster_yale.pdf).

[57] LIght Weight WEka COmpatible Data Mining Pre-processing ToOL https://sourceforge.net/projects/liwecool/