Discovery and Evaluation of Student's Profiles with Machine Learning

Evis Trandafili
Faculty of Information Technology
Polytechnic University of Tirana
Tirana, Albania
etrandafili@fti.edu.al

Alban Allkoçi
Faculty of Information Technology
Polytechnic University of Tirana
Tirana, Albania
aallkoci@fti.edu.al

Elinda Kajo
Faculty of Information Technology
Polytechnic University of Tirana
Tirana, Albania
ekajo@fti.edu.al

Aleksandër Xhuvani
Faculty of Information Technology
Polytechnic University of Tirana
Tirana, Albania
axhuvani@fti.edu.al

ABSTRACT

Higher education institutions are overwhelmed with huge amounts of information regarding student's enrollment, number of courses completed, achievement in each course, performance indicators and other data. This has led to an increasingly complex analysis process of the growing volume of data and to the incapability to take decisions regarding curricula reform and restructuring. On the other side, educational data mining is a growing field aiming at discovering knowledge from student's data in order to thoroughly understand the learning process and take appropriate actions to improve the student's performance and the quality of the courses delivery. This paper presents a thorough analysis process performed on student's data through machine learning techniques. Experiments performed on a very large real-world dataset of students performance on all courses of a university, reveal interesting and important students profiles with clustering and surprising relationships among the courses performance with association rule mining.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Clustering*

General Terms

Experimentation

Keywords

Data mining, knowledge discovery, clustering, student profile, machine learning

1. INTRODUCTION

Recently, a growing amount of data is being continuously gathered at universities regarding student's performance, their interactions with various actors and systems at the institution and other related indicators. The goal is analyzing data for decision making towards reform and restructuring of courses delivery improvement of teaching quality, and enhancement of learning capabilities of students. In this context, a growing field such as Educational Data Mining (EDM) aims at exploiting data coming from learning settings in educational environments in order to better understand the learning process and the student profile [1]. In this research area, intensive efforts are being dedicated to using automatic methods to infer knowledge from the student's data. Methods already established in the machine learning and data mining field, successfully applied to many problems [2], have the potential to discover interesting patterns in the collected data regarding assessment components of courses and various interaction indicators related with the students life within a university.

Some successful applications of machine learning algorithms have produced interesting results. However, there is much open research due to the many challenges that arise for this kind of problem. First of all, data gathered at universities were not initially planned for decision making analysis and this gives rise to several different data formats or schemas which are hard to handle and process with existing mining techniques. Therefore, it is of critical importance to perform data engineering and preprocessing such that the data input is appropriate for machine learning methods. In addition, operations such as data cleaning and missing values completion are essential in order to produce accurate and meaningful models of the student's performance.

The research presented in this paper is an ongoing effort at the Polytechnic University of Tirana (UPT) to exploit collected data about student's performance for the purpose of planning curricular revisions and restructuring the course delivery process. In particular, data analysis aims at the following: evaluating the quality of courses delivery; understanding and assessing strengths and weaknesses of students in particular subjects; discovering anomalies in the teaching and learning process; identifying potential improvements in the university infrastructure and other general interventions in the overall learning environment.

In this paper, we perform a deep analysis on the performance of students at UPT in three different bachelor programs at the Faculty of Information Technology: Informatics, Electronics and Telecommunications. We analyze a large dataset of student's results comprising their evaluations for all the subjects. With expectation-maximization clustering, we discover several important profiles that reveal some interesting patterns of students. This partitioning leads to important suggestions for improvement in the delivery of some courses. Furthermore, we perform association rule mining, with the goal of discovering patterns in the performance of students regarding different subjects. The discovered rules imply interesting relations among particular subjects where the performance of students is strongly related. This picture also leads to a better understanding of the student's strength or weakness in some groups of subjects.

This paper is structured as follows: Section 2 presents related work, Section 3 presents the preprocessing and input engineering steps, Section 4 presents experiments and discussion of results and we conclude in Section 5.

BCI'12, September 16–20, 2012, Novi Sad, Serbia.

2. RELATED WORK

In this section we review some recent research works regarding the use of machine learning and data mining for analyzing educational data.

In [3] the authors present an approach for comparing several classification methods for the purpose of predicting the final outcome of a course. The datasets used in this work are relatively small, 125 and 88 rows respectively. They employ for numerical data, classifiers based on multiple linear regression and support vector machines and for categorical data three variations of Naïve Bayes classifier.

Another recent interesting approach was presented in [4]. In this work the author aimed at identifying up to what extent the enrolment data can be used to predict student's success. He applied decision tree algorithms to student enrollment data taken from an information system of a university and generated two decision trees to classify successful and unsuccessful students.

Another problem tackled by recent research is predicting which students are likely to continue their education with the postgraduate degree. In [5] the authors analyze data collected through surveys from senior undergraduate students and apply decision tree algorithms in the Weka tool [6] such as ID3. The reported results are very promising showing a classification accuracy of 88%.

In [7] the authors deal with the problem of predicting the performance of students in future courses given the performance in previous ones. The proposed approach is based on decision tree algorithms and is applied on engineering students' past performance data to generate the model and this model is then used to predict the students' performance. The goal of the authors is to identify in advance the students who are likely to fail in particular subjects and allow the teacher to handle appropriately these cases. The presented results are very positive and the authors show that the accuracy of the model may increase with more data coming for further results of the students.

One problem tightly related with the students profile is the analysis of interaction behaviors. In [8], the authors present a modeling framework that uses both unsupervised and supervised classification to build student models for exploratory learning environments. They apply the framework to build student models for two different learning environments and using two different data sources (logged interface and eye-tracking data). Their dataset was rather limited but however, the authors managed to provide evidence that the framework can automatically identify meaningful student interaction behaviors and can be used to build user models for the online classification of new student behaviors. In addition, they show that the framework can be transferred across applications and data types.

One approach is exploiting data coming from learning management system such as Moodle. In [9] the authors propose the use of a special type of association rules mining for discovering interesting relationships from the students ' test data collected with Moodle. They apply Class Association Rule (CAR) mining to different data matrices such as the score-matrix, the relationship-matrix and the knowledgematrix. These matrices are constructed based on the data related to students' performance in the test and on the domain knowledge provided by the instructor. The authors show how to obtain the matrix and how to apply the CAR learning algorithm.

Another interesting work has been recently presented in [10] where the authors propose a Two-Phase Fuzzy Mining and Learning Algorithm, integrating data mining algorithm, fuzzy set theory, and the look-ahead mechanism, to find the embedded information. They show how this approach can be provided to teachers for further analyzing, refining or reorganizing the teaching materials and tests, from historical learning records.

An important problem related with educational systems is e-learning. In [11] it was presented a work on the hybridization of artificial intelligence techniques and statistical tools to evaluate and adapt the e-learning systems including e-learners. In the e-learning process, the learner's profile plays a crucial role in the evaluation process and the recommendations to improve the overall process. The author in this work proposed a classification of learners into specific categories based on the learner's profiles; the learners' classes named as regular, workers, casual, bad, and absent. The approach exploits extracted statistical usage patterns that give a clear map describing the data and helping in constructing the e-learning system. One of the main goals of the work is to find out how to make the irregular students who are away back to be regular ones and also find a method to evaluate the e-learners as well as to adapt the content and structure of the e-learning system. From an algorithmic point of view, the work introduces the application of different fuzzy clustering techniques to find the learners profiles. The experimental results show that there is a match with a 78% with the real world behavior and that the fuzzy clustering reflects the learners' behavior perfectly.

3. PREPROCESSING AND INPUT ENGINEERING

3.1 Input Format

The data was collected joining various tables of the student's database from the information system (Figure 1) of the Polytechnic University of Tirana (PUT) and exported into a MySQL database. In this table we have for every student all the subjects in which the student is enrolled and the relative evaluation. The data regard all students and all subjects for three engineering programs such as Informatics, Electronics and Telecommunication in the Faculty of Information Technology. Each row of the following .xlsx file summarizes the integral parts of one course for the specified student. This table initially has approximately 35 000 rows to be exported into MysSQL database shown in Figure 2 where we can easily notice some missing values. These values correspond to the students who have not participated in the final exam of this course for various reasons, (they have not succeeded in the lab, the final project or they have attended less than 75% of the course) [12], therefore they do not have an assessment yet.

	A	B	C	D	E	F	G	H	I	J
1	ID	ID_Course	Course Name	ID_STUDENT	Name	Surname	Grade	Presence	Project	Lab
2	202	L110	C1	BE0001	Name here	Surname here	6	100	1	1
3	1032	L107	C2	BE0001	Name here	Surname here	5	100	1	1
4	1485	L101	C3	BE0001	Name here	Surname here	9	100	1	1
5	1714	L112	C4	BE0001	Name here	Surname here	9	100	1	1
6	1893	L103	C5	BE0001	Name here	Surname here	7	100	1	1
7	2792	L102	C6	BE0001	Name here	Surname here	6	100	1	1
8	2976	L108	C7	BE0001	Name here	Surname here	4	100	1	1
9	3227	L104	C8	BE0001	Name here	Surname here	8	100	1	1
10	3412	L111	C9	BE0001	Name here	Surname here	5	100	1	1
11	5201	L113	C10	BE0001	Name here	Surname here	6	100	1	1
12	5425	L105	C11	BE0001	Name here	Surname here	7	100	1	1
13	5660	L106	C12	BE0001	Name here	Surname here	5	100	1	1
14	7446	L109	C13	BE0001	Name here	Surname here	5	100	1	1
15	17580	L210	C14	BE0001	Name here	Surname here	7	100	1	1
16	19003	L201	C15	BE0001	Name here	Surname here	10	100	1	1
17	19666	L203	C16	BE0001	Name here	Surname here	6	100	1	1
18	21558	L207	C17	BE0001	Name here	Surname here	6	100	1	1
19	22648	L202	C18	BE0001	Name here	Surname here	6	100	1	1
20	26174	L206	C19	BE0001	Name here	Surname here	6	100	1	1
21	27933	L209	C20	BE0001	Name here	Surname here	5	88	1	1
22	28319	L213	C21	BE0001	Name here	Surname here	8	85	1	1
23	30912	L204	C22	BE0001	Name here	Surname here	5	100	1	1

Figure 1: The data collected from the information system of the university.

	ID	ID_LENDA	ID_STUDENT	VLERESIMI	FREKUENTIMI	DETYRA	LABORATORE
▸	1	L110	BE0434	5	100	1	1
	2	L110	BE0435	5	100	1	1
	3	L110	BE0436	5	100	1	1
	4	L113	BE0233		100	1	1
	5	L113	BE0239		100	1	1
	6	L113	BE0499		0	1	1
	7	L113	BE0506	5	100	1	1
	8	L113	BE0352	5	100	1	1
	9	L113	BE0358	5	100	1	1
	10	L113	BE0369	9	100	1	1

Figure 2: The data exported into MySQL database.

3.2 Data Transformation

In order to prepare the input for the Weka software, we need to generate a file with the extension .arff compatible with the requirements of the Weka program input. In this context we need one single record for each student, whose attributes are all the courses for that particular bachelor degree program. As we have to deal with three different programs, consequently with three different groups of attributes, we need to generate one .arff file for each program. For this reason we have developed a program in the Java language that translates the data into the required format of Weka according to the program where the student is enrolled.

It is crucial to notice here that the number of missing values has become greater, due to the fact that we had to join the syllabus of the program with the courses that the student has completed. So we expect to have missing values for example for the students frequenting the first year of studies and have not completed yet the courses of the second and third year of his studies.

The above transformation applied to the data stored in MySQL database, resulted in three .arff files, one for each program of study. Figure 3 shows a portion of the attributes listed in the .arff file of the Informatics program, which start with the student ID attribute followed by the assessment, attendance, project and laboratory attribute for each course. Here the attributes shown regard the course with the code L111. As a result, the real .arff file includes 1 attribute to identify the student and 148 attributes that correspond to the evaluations of 37 courses.

```
@relation student
@attribute 'ID' string
@attribute 'L111_VLERESIMI' real
@attribute 'L111_FREKUENTIMI' real
@attribute 'L111_DETYRA' real
@attribute 'L111_LABORATORE' real
```

Figure 3: Part of the attributes section on the .arff file.

The content of the .arff file's data regarding the record of one single student of the informatics program is shown in Figure 4. Each value matches the attributes listed in Figure 3. The overall content of the .arff file contains the data regarding 485 students enrolled in the Informatics program.

```
@data
BI0181,5,100,1,1,5,100,1,1,5,100,1,1,5,100,1,1,5,100,1,1,5,100,1
,1,5,100,1,1,6,100,1,1,6,100,1,1,5,100,1,1,5,100,1,1,10,100,1,1,
?,100,1,1,5,100,1,1,8,100,1,1,6,100,1,1,7,100,1,1,6,100,1,1,5,75
,1,1,?,100,0,1,4,100,1,1,5,100,1,1,5,100,1,1,?,93,0,1,?,80,1,1,5
,100,1,1,7,100,1,1,?,85,0,0,5,100,1,1,5,100,1,1,3,100,1,1,?,95,1
,1,?,100,0,1,10,100,1,1,?,100,1,1,9,100,1,1,?,100,0,1
```

Figure 4: Student record in .arff file.

Moreover, this pattern also applies to the .arff file of the Electronic and Telecommunication programs with different courses and/or number of courses according to the syllabus of each program. In particular, the .arff file of the Electronics program contains information regarding 41 courses and 510 students enrolled and the .arff file of the Telecommunication program contains information regarding 37 courses and 435 students enrolled.

3.3 Data Cleansing and Preparation

One of the main breakdowns of the university's data is that these data come with noise such as data insertion errors and missing or incorrect values. In order to be able to analyze such data with machine learning algorithms such as association rule mining, clustering or decision trees, missing values should be handled properly. For this reason we replaced missing values with the filter weka.filters.unsupervised.attribute.ReplaceMissingValues. The algorithm is run only on final assessment.

In particular, for the evaluation results in each course of the program where the student is enrolled, the used filter replaces the missing values with the mean value of the matching attribute. As mentioned earlier in this paper, the number of missing values is not insignificant according to the numbers. So, we will have to deal with the replaced data when analyzing the experimental results.

In addition, an unsupervised discretization procedure has been applied to all the numeric attributes such that these attributes can be handled by the learning algorithm such as Apriori [13], using weka.filters.unsupervised.attribute.Discretize. The algorithm run to the student's final assessment with a predefined number of bins equals to 4. This resulted in identifying 4 distinct intervals as listed in Figure 5. Analyzing these results we can notice that the intervals generated are meaningful to the fact that the algorithm was applied to the assessment attribute which ranges through integer values between 4 and 10. The grades can naturally be grouped by the student's performance and we can interpret them as low, good, very good and excellent.

No.	Label	Count
1	'(-inf-5.5]'	203
2	'(5.5-7]'	190
3	'(7-8.5]'	60
4	'(8.5-inf)'	66

Figure 5: The result of the discretization procedure.

4. EXPERIMENTS

In this section we present the experiments with clustering and association rule mining on the three preprocessed datasets of the university, respectively the .arff file of the Informatics program, the .arff file of the Electronics program and the .arff file of the Telecommunication program. The experiments ran in Weka using all the attributes that express the final results of the courses, and setting aside the attributes of attendance, project and laboratory.

4.1 Models Generated

The clustering algorithm used is expectation-maximization (EM) [13] clustering as implemented in Weka. We performed two experiments: one where the algorithm is left to find the number of clusters automatically and one other where the number of clusters is given as input parameter to the algorithm. In the second experiment we used as number of clusters k=4, while in the first case the algorithm run with default parameters and found the optimal k=7. Figure 6 shows the clusters generated in the first case and Figure 7 shows the clusters generated in the second experiment. The cluster results' are shown for a single course evaluation, even if the output file repeats clustering for each course.

Number of clusters selected by cross validation: 7

Attribute	Cluster 0 (0.16)	1 (0.26)	2 (0.11)	3 (0.06)	4 (0.28)	5 (0.04)	6 (0.09)
L110_VLERESIMI							
'(-inf-5.5]'	53.9827	66.7131	26.9925	2	18.9846	1	40.3271
'(5.5-7]'	29.862	53.868	21.435	4.2709	71.5616	6.0023	10.0003
'(7-8.5]'	2.0024	10.5808	7.1136	7.5037	29.5721	9.2274	1
'(8.5-inf)'	1.0193	5.3493	6.0059	19.5867	29.8344	10.2042	1
[total]	86.8664	136.5112	61.547	33.3614	149.9527	26.4339	52.3274

Figure 6: The clusters found automatically.

Number of clusters: 4

Attribute	Cluster 0 (0.56)	1 (0.22)	2 (0.11)	3 (0.11)
L110_VLERESIMI				
'(-inf-5.5]'	124.8505	72.7674	7.359	2.0231
'(5.5-7]'	122.1009	34.91	26.2805	10.7086
'(7-8.5]'	27.198	5.798	13.3165	17.6874
'(8.5-inf)'	18.2086	6.0456	15.7867	29.9591
[total]	292.3579	119.5211	62.7427	60.3782

Figure 7: The clusters found with input parameters.

The other group of experiments focused on rules and relations between various courses of the programs where the students are enrolled. In Figure 8 is shown the result of the experiment ran on Informatics program preprocessed data. We used the Apriori algorithm with input parameters the number of best rules found. In the same way, the experiment was repeated for the three programs with different parameters for the best rules.

1. L305_VLERESIMI='(6.4-7]' L311_VLERESIMI='(7.5-8]' 414 ==> L312_VLERESIMI='(7.4-7.8]' 410 conf:(0.99)
2. L311_VLERESIMI='(7.5-8]' L312_VLERESIMI='(7.4-7.8]' 418 ==> L305_VLERESIMI='(6.4-7]' 410 conf:(0.98)
3. L305_VLERESIMI='(6.4-7]' 433 ==> L312_VLERESIMI='(7.4-7.8]' 424 conf:(0.98)
4. L312_VLERESIMI='(7.4-7.8]' 438 ==> L305_VLERESIMI='(6.4-7]' 424 conf:(0.97)
5. L305_VLERESIMI='(6.4-7]' L312_VLERESIMI='(7.4-7.8]' 424 ==> L311_VLERESIMI='(7.5-8]' 410 conf:(0.97)
6. L311_VLERESIMI='(7.5-8]' 433 ==> L312_VLERESIMI='(7.4-7.8]' 418 conf:(0.97)
7. L311_VLERESIMI='(7.5-8]' 433 ==> L305_VLERESIMI='(6.4-7]' 414 conf:(0.96)
8. L305_VLERESIMI='(6.4-7]' 433 ==> L311_VLERESIMI='(7.5-8]' 414 conf:(0.96)
9. L312_VLERESIMI='(7.4-7.8]' 438 ==> L311_VLERESIMI='(7.5-8]' 418 conf:(0.95)
10. L311_VLERESIMI='(7.5-8]' 433 ==> L305_VLERESIMI='(6.4-7]' L312_VLERESIMI='(7.4-7.8]' 410 conf:(0.95)

Figure 8: Best rules found by Apriori algorithm.

Another approach to evaluate students' performance and predict future course assessments is to build a prediction model based on the previous and current courses results. This gives the possibility to predict in advance for each student, the potential of failure in a certain course giving therefore the chance to handle this risk of failure appropriately before the examination period starts. In order to build such model, we need to exploit the grades of the student in the ongoing year in addition to the grades of the previous years.

One prediction model that has proven to be very successful is decision tree classification [2]. This model is build from training data and has very good capability to predict classes with nominal attributes. In our example, we would like to exploit the previous results of the student, to predict whether she will fail or pass a certain course that is to be examined in the next coming examination session. In order to learn decision trees we first need to generate the dataset in an appropriate format. We have followed the same steps as per the previous experiments and the pre-processing tasks as explained in the previous section. Since decision trees need nominal attributes, we have considered every set of course evaluation as a discrete set of 7 values, i.e., 4 is the value for the fail grade and 10 the value for the maximum grade. In order to perform the experiments, we used the J48 algorithm implemented in Weka. In addition, we have applied the replace missing values procedure as built in Weka software. We have performed two experiments with decision tree learning. In the first experiment we considered only the course evaluation of the third year, while in the second experiment we added also the course evaluations of the second year of studies, in each case using the same group of students. In both experiments we used binary splits of the decision tree because the result was very difficult to interpret due to the large amount of data. The binary splits are applied to the nominal attributes. In the second experiment we noticed that the tree was very compact so we repeated the experiment with non binary splits, in order to have more information about the results.

The generated model for the first experiment is shown in Figure 9 and Figure 10. The decision tree is large and we only show it partially.

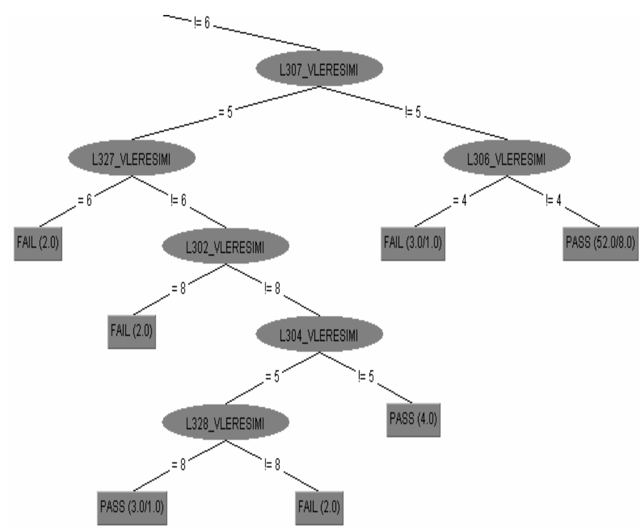

Figure 9: Partial decision tree with only third year courses.

```
J48 pruned tree
------------------

L305_VLERESIMI = 7
|   L308_VLERESIMI = 4: FAIL (2.0)
|   L308_VLERESIMI != 4
|   |   L306_VLERESIMI = 6
|   |   |   L302_VLERESIMI = 10
|   |   |   |   L327_VLERESIMI = 8: FAIL (2.0)
|   |   |   |   L327_VLERESIMI != 8: PASS (2.0)
|   |   |   L302_VLERESIMI != 10: FAIL (5.0)
|   |   L306_VLERESIMI != 6
|   |   |   L307_VLERESIMI = 5
|   |   |   |   L327_VLERESIMI = 6: FAIL (2.0)
|   |   |   |   L327_VLERESIMI != 6
|   |   |   |   |   L302_VLERESIMI = 8: FAIL (2.0)
|   |   |   |   |   L302_VLERESIMI != 8
|   |   |   |   |   |   L304_VLERESIMI = 5
|   |   |   |   |   |   |   L328_VLERESIMI = 8: PASS (3.0/1.0)
|   |   |   |   |   |   |   L328_VLERESIMI != 8: FAIL (2.0)
|   |   |   |   |   |   L304_VLERESIMI != 5: PASS (4.0)
|   |   |   L307_VLERESIMI != 5
|   |   |   |   L306_VLERESIMI = 4: FAIL (3.0/1.0)
|   |   |   |   L306_VLERESIMI != 4: PASS (52.0/8.0)
L305_VLERESIMI != 7: PASS (52.0/1.0)

Number of Leaves  :     12

Size of the tree :      23
```

Figure 10: Decision tree generated with third year courses.

In the second experiment we considered also the second year course evaluations. The tree generated in this case is smaller and more compact as shown in Figure 11 where we show it as a whole.

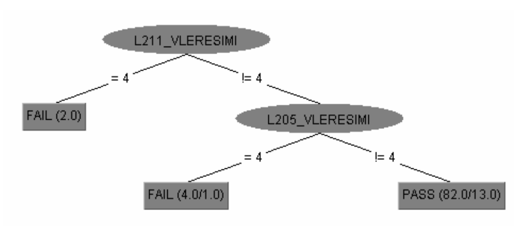

Figure 11: Binary decision tree generated with courses of second and third year.

```
L312_VLERESIMI = 4: PASS (0.0)
L312_VLERESIMI = 5: PASS (2.0)
L312_VLERESIMI = 6: PASS (3.0)
L312_VLERESIMI = 7: PASS (10.0)
L312_VLERESIMI = 8
|   L305_VLERESIMI = 4: PASS (2.0)
|   L305_VLERESIMI = 5: PASS (3.0)
|   L305_VLERESIMI = 6: PASS (4.0/1.0)
|   L305_VLERESIMI = 7
|   |   L201_VLERESIMI = 4: PASS (0.0)
|   |   L201_VLERESIMI = 5: PASS (0.0)
|   |   L201_VLERESIMI = 6: PASS (0.0)
|   |   L201_VLERESIMI = 7: PASS (0.0)
|   |   L201_VLERESIMI = 8: PASS (6.0)
|   |   L201_VLERESIMI = 9
|   |   |   L205_VLERESIMI = 4: FAIL (1.0)
|   |   |   L205_VLERESIMI = 5: PASS (4.0)
|   |   |   L205_VLERESIMI = 6: PASS (1.0)
|   |   |   L205_VLERESIMI = 7: PASS (2.0)
|   |   |   L205_VLERESIMI = 8: PASS (0.0)
|   |   |   L205_VLERESIMI = 9: PASS (0.0)
|   |   |   L205_VLERESIMI = 10: PASS (0.0)
|   |   L201_VLERESIMI = 10: PASS (35.0/16.0)
|   L305_VLERESIMI = 8: PASS (4.0)
|   L305_VLERESIMI = 9: PASS (1.0)
|   L305_VLERESIMI = 10: PASS (2.0)
L312_VLERESIMI = 9: PASS (8.0)
L312_VLERESIMI = 10: PASS (0.0)

Number of Leaves  :     25

Size of the tree :      29
```

Figure 12: Non-Binary decision tree with the second and the third year courses, showing pass and fail results.

Due to the size of the tree in Figure 11, we had the opportunity to repeat the experiment with the binary splits removed. The decision tree generated is shown in Figure 12. Even if it is more large than the previous one, it is still not very complex to understand and analyze.

4.2 Evaluation of Results

Our qualitative investigation of the clusters, tried to group the big amount of student data into some logically significant categories. The first experiment of clustering that ran using expectation-maximization algorithm with automatic find of the number of clusters, resulted in 7 clusters. Analyzing the containing data into each cluster needs some background information about the preprocessing of that data. In Figure 6 we have shown some results of clustering with automatic number of clusters. It matches the clustering of the course L110. The same situation is obtained for all the courses in the syllabus. Clustering is applied in the overall data of the program, so the clusters identify various categories of students according to their enrollment in all the courses. The process of analyzing the results of 7 clusters turns to be difficult if any prior knowledge is applied. The first important hint that captures our attention is the maximum total number of students that fall in cluster 4 which is immediately followed by cluster 1. Every other cluster seems to differ significantly with cluster 1 and 4, according to the number of students that fall in it. If we focus on these two dominant clusters we notice that cluster 4 can be named “very good and excellent” and cluster 1 can be named “low and good”. We also notice that a normal distribution would have generated a cluster with a maximum total of students falling in the average student’s grade. This is not our case and yields to a difficult to answer situation which should be clarified with other sources of knowledge. We evaluated the situation and found out that the course shown here has a prerequisite course; as a result every student that failed in the prerequisite course does not take the exam of L110. During the preprocessing phase we replaced missing values with Weka filter which replaced missing values with the mean value. Furthermore, usually good and very good students who are ambitious about having very good grades deliberately get failed if they feel they can’t get a good assessment during the final exam. In this way they have the right to take the exam during the next exam session and get a better grade. This fact explains the big number of low grades including the fail ones and explains the deviation from the normal distribution of the student’s grades. Going back to Figure 6 we notice that cluster 3 and cluster 5 do not differ much from each other, so this clustering can be wrapped into a single cluster. The same evidence is shown for cluster 0 and cluster 2. The pattern of closeness among clusters leads to the next experiment with a custom number of clusters. Since the evaluation of automatic clusters resulted in 7 clusters relatively difficult to analyze, we did the next experiment as shown in Figure 7, running EM algorithm with the number of result clusters set to 4. According to the results in Figure 7, we have 4 compressed clusters. The data here is easier to understand and to analyze the meaning of clusters. The cluster with the excellent students is cluster 3 which includes 11% of the students. Cluster 0 shows the profile of students who are not very probable of continuing the master studies. This cluster has 56% of the overall students. Here the percentage of students with low grades seems to be very high. A more careful evaluation should be done from the curriculum designers which can lead to further insights on students’ difficulties. This observation is clearer if we explore all the result set with all the courses. As a conclusion of applying clustering over student’s datasets, we determine that a lot of students have problems with various courses and only 11% of them manage to get very good results. This can be used as a statistical model to evaluate the number of students that will continue master studies in the university. In this way the academic staff can be prepared for the new academic year.

The next experiment used the Apriori algorithm. The output rules are shown in Figure 8. Rule number 2 shows us an interesting point explained as follows: the students getting 8 in the course L311 and also 8 in course L312, are likely to get 7 in the course L305. Following the same reasoning with the rule number 6, we can predict the assessment being 8 for the course L312 if we previously know that the assessment of the course L311 is also 8. At this point, it is interesting to mention that L311 is the code for the internship and the code L312 corresponds to the diploma thesis. This rule is noticed even within the three different programs were students are enrolled. In this context the university should try to integrate the diploma thesis with the internship, because of the relevant results obtained here which are based on an accuracy of 97% of the Apriori algorithm.

Another interesting rule is extracted from the experiment ran with greater number of rules. The accuracy was slightly smaller but the big amount of input data favor the precise interpretation. The rule that got our attention was the following:

```
L305_VLERESIMI='(6.4-7]' L303_VLERESIMI='(5.5-
6.4]' L311_VLERESIMI='(7.5-8]'
L301_VLERESIMI='(5.8-6.4]' 367 ==>
L312_VLERESIMI='(7.4-7.8]' 367    conf:(1)
```

Figure 13: The rule that explains the relationships with the diploma thesis.

L312, as mentioned above, is the evaluation for the diploma thesis. This rule shows an interesting relationship with a group of courses matching the following: Operating Systems, Object Oriented Programming, Electronics and Internship.

Regarding the results obtained with decision trees, in the first experiment, the classification accuracy is 79.38% while in the second experiment, with courses from second and third year, the accuracy is 82%. This shows that using more information about the previous student performance, leads to better results in predicting whether a student will fail or pass in a certain course in the next examination session.

5. CONCLUSION

In this paper we have presented an approach for discovering student profiles from course evaluation data and for generating associations between subjects based on the student performance. We have employed expectation-maximization clustering to partition students in distinct profiles that show some interesting features when analyzed qualitatively. In addition, we have shown through association rule learning that it is possible to discover interesting relationships among the different subjects of a student. Furthermore, we applied decision trees' algorithms to predict student assessment. The experiments' results show that we can predict in advance the risk of a student to fail in a certain course based on his performance in the previous courses.

Experiments performed on a very large real-world dataset of a university demonstrate the effectiveness of the approach with the goal of improving the university curricula, course delivery and general teaching infrastructure and related services.

As a future work we will work on applying some other data mining algorithms in processing more information about the students' progress in each course in order to optimize the results.

6. REFERENCES

[1] Romero, Cr. and Ventura, S., 2010. Educational Data Mining: A Review of the State of the Art. *In IEEE Transactions on Systems, Man, and Cybernetics* – Part C: Applications and Reviews; 40/6: 601–18.

[2] Witten, I.H., Frank, E. and Hall, M.E., 2011. *Data Mining: Practical Machine Learning Tools and Techniques.* Third edition, Morgan Kaufmann.

[3] Hamalainen, W. and Vinni, M. 2006. Comparison of machine learning methods for intelligent tutoring systems. *In Mitsuru Ikeda, Kevin Ashley, and Tak-Wai Chan, editors, Intelligent Tutoring Systems*, volume 4053 of Lecture Notes in Computer Science, pages 525–534. Springer Berlin / Heidelberg.

[4] Kovacic, Z. J. 2010. Early prediction of student success: Mining student enrollment data. In *Proceedings of Informing Science & IT Education Conference (InSITE)*.

[5] Bresfelean, V. P. 2007. Analysis and Predictions on Students' Behavior Using Decision Trees in Weka Environment. In *Proceedings of the ITI 2007 29th Int. Conf. on Information Technology Interfaces*, June 25–28.

[6] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten I.H. 2009. *The WEKA Data Mining Software: An Update*. SIGKDD Explorations, Volume 11, Issue 1.

[7] Kabra, R. R. and Bichkar, R. S. Performance Prediction of Engineering Students using Decision Trees in *International Journal of Computer Applications* (0975–8887).

[8] Amershi, S. and Conati, C., 2009. Combining Unsupervised and Supervised Classification to Build User Models for Exploratory Learning Environments in *Journal of Educational Data Mining*, 1(1), 18–71.

[9] Romero, C., Ventura, S., Vasilyeva, E. and Pechenizkiy, M., 2010. Class Association Rules Mining from Students' Test Data. In Baker, R.S.J.d., Merceron, A., Pavlik, P.I. Jr. (Eds.) Proceedings of the 3rd Int. Conf. in *Educational Data Mining* (pp. 317-318). Pittsburgh, USA.

[10] Tsai, C. J., Tseng, S. S., and Lin, C. Y., 2001. A Two-Phase Fuzzy Mining and Learning Algorithm for Adaptive Learning Environment. In *ICCS '01: Proceedings of the Int. Conf. on Computational Science-Part II*, (pp. 429–438). London, UK: Springer-Verlag.

[11] Agrawal, R. and Ramakrishnan, S., 1994. Fast algorithms for mining association rules in large databases. Proc. of the 20th International Conference on Very Large Data Bases, VLDB, pages 487–499, Santiago, Chile, September.

[12] Volume 36– No.11, December 2011. Academic Regulation of the Polytechnic University of Tirana. http://www.fti.edu.al/data/pages/18/attach/Rregullore%20Mesimore_UPT.pdf

[13] Mclachlan, G.J. and Krishnan, J., 1996. *The EM Algorithm and Extensions in Wiley-Interscience*, First edition.