Statistical & Machine Learning Techniques in Adaptive Learning Environments

Master Thesis
Chatzistavrou Kyriaki, RN:548

Thesis Committee:
Supervisor: Lefteris Angelis, Professor
Ioannis Stamelos, Professor
Christina Boutsouki, Associate Professor

THESSALONIKI
FEBRUARY 2017
Acknowledgements

This work was supported by the EU Erasmus+ project entitled “Competence Oriented Multilingual Adaptive Language Assessment and Training System” (http://www.comalat.eu). I wish to thank my partners and especially, my supervisor professor Dr. Lefteris Angelis for not only spending his valuable time reading my chapters, but for also offering invaluable insight that significantly improved the final product.

Particularly, I would like to thank Dr. Nikolaos Mittas for the collaboration on fuzzy grading, especially for conducting the validation of the fuzzy grading algorithm as reported in the Section 3.8 and Appendix B – Results of Testing.
Abstract

Due to increasing global competition and technology’s constant progress, the usage of online training platforms has grown rapidly. In this thesis, we present statistical & machine learning techniques that can be used in Adaptive Learning Environments. More specifically, we present a grading technique based on fuzzy numbers, which reduces the problem that most platforms face; the lack of a physical evaluator. Moreover, we present an effective way to motivate learners to continuously try and improve their performance by showing them their ranking among the other learners. Interesting rules could be discovered using association rules mining, which could help the learning procedure and the creation of material that reflects the learners’ overall interests or needs.

Keywords: “Statistics, Fuzzy grading, Membership functions, Evaluation method, Adaptability, Learning platforms, Training Platforms, Association Rules, Shiny, R, arules, arulezViz, Visualization Techniques”

Kyriaki Chatzistavrou
26/01/2017
# Contents

ACKNOWLEDGEMENTS ...................................................................................... I

ABSTRACT ........................................................................................................ II

CONTENTS ....................................................................................................... III

TABLE OF TABLES ........................................................................................... VI

TABLE OF FIGURES ......................................................................................... VII

1 INTRODUCTION ............................................................................................. 9

2 THE COMALAT PLATFORM ............................................................................ 11
   2.1 COMALAT TOOLS ...................................................................................... 11
   2.2 ROLE IN COMALAT PROJECT .................................................................. 13
   2.3 COMALAT STRUCTURE ........................................................................... 14
   2.4 USER TRACKING INFORMATION IN COMALAT ........................................ 15

3 FUZZY GRADING ............................................................................................ 17
   3.1 INTRODUCTION TO FUZZY LOGIC ......................................................... 17
      3.1.1 Fuzzy logic system ............................................................................. 17
      3.1.2 Linguistic Variables ........................................................................... 18
      3.1.3 Membership Functions ..................................................................... 19
      3.1.4 Fuzzy Rules ...................................................................................... 19
      3.1.5 Fuzzy Set Operations ........................................................................ 20
      3.1.6 Defuzzification .................................................................................. 20
   3.2 INTRODUCTION TO FUZZY GRADING .................................................... 20
   3.3 RELATED WORK ....................................................................................... 21
   3.4 IMPLEMENTATION OF LAW’S METHOD IN FUZZY GRADING WITH R
      PROGRAMMING LANGUAGE ....................................................................... 23
   3.5 USAGE OF FUZZY GRADING AND EXAMPLES ........................................ 27
   3.6 FUZZY GRADING WEB APPLICATION ..................................................... 29
   3.7 PREDETERMINED PERCENTAGES OF FUZZY GRADING ....................... 30
3.8 Testing .............................................................................................................. 31

4 RANKING ............................................................................................................. 35

5 ASSOCIATION RULES MINING ......................................................................... 39

  5.1 INTRODUCTION ............................................................................................... 39
  5.2 THEORETICAL BACKGROUND ........................................................................ 40
  5.3 PACKAGES ......................................................................................................... 43
    5.3.1 arules ............................................................................................................ 44
    5.3.2 arulesViz ....................................................................................................... 47
  5.4 THE METHOD ..................................................................................................... 52

6 CONCLUSION ......................................................................................................... 59

BIBLIOGRAPHY ......................................................................................................... 61

APPENDIX A – FUZZY GRADING APPLICATION .................................................... 65

APPENDIX B – RESULTS OF TESTING .................................................................. 84

  CASE STUDY 1 ........................................................................................................ 84
  CASE STUDY 2 ........................................................................................................ 85
  CASE STUDY 3 ........................................................................................................ 86
  CASE STUDY 4 ........................................................................................................ 87
  CASE STUDY 5 ........................................................................................................ 88
  CASE STUDY 6 ........................................................................................................ 89
  CASE STUDY 7 ........................................................................................................ 90
  CASE STUDY 8 ........................................................................................................ 91
  CASE STUDY 9 ........................................................................................................ 92
  CASE STUDY 10 ..................................................................................................... 93
  CASE STUDY 11 ..................................................................................................... 94
  CASE STUDY 12 ..................................................................................................... 95
  CASE STUDY 13 ..................................................................................................... 96
  CASE STUDY 14 ..................................................................................................... 97
  CASE STUDY 15 ..................................................................................................... 98
  CASE STUDY 16 ..................................................................................................... 99

APPENDIX C – RANKING APPLICATION ................................................................. 100
APPENDIX D – ASSOCIATION RULES MINING

INPUT MATRIX .................................................................................................................. 105
ASSOCIATION RULES MINING CODE .............................................................................. 109
GLOSSARY .......................................................................................................................... 112
Table of Tables

Table 1 – Key pieces of information ............................................................... 16
Table 2 – Fuzzy Rules Example ...................................................................... 19
Table 3 – Example Fuzzy Set Operations .......................................................... 20
Table 4 - The raw scores of 10 learners and their corresponding grade .......... 28
Table 5 - Using Fuzzy Grading in extreme cases ............................................. 28
Table 6 – Simulated dataset ........................................................................... 32
Table 7 – Alternative scenarios for ideal percentages ...................................... 33
Table 8 – An example supermarket database with 5 transactions .................. 41
Table 9 – An example of a collection of supermarket data presented as a binary incidence matrix ......................................................................................... 41
Table 10 – Attributes used for each learner ..................................................... 52
Table 11 – Example input matrix .................................................................... 54
Table 12 – Example sparse matrix ................................................................... 54
Table of Figures

Figure 1 – COMALAT Guide Tool Profile Page .......................................................... 12
Figure 2 – Language Structure .................................................................................. 15
Figure 3 – Section Structure ...................................................................................... 15
Figure 4 – Describe Temperature with fuzzy logic ..................................................... 17
Figure 5 – Example of a Fuzzy Logic System .............................................................. 18
Figure 6 – Different types of membership functions .................................................... 19
Figure 7 – Fuzzy Grading Web Application ............................................................... 30
Figure 8 – The ranking web application .................................................................... 35
Figure 9 – Ranking application input-output (non-graphical) area ............................... 36
Figure 10 – Ranking App: Histograms ..................................................................... 37
Figure 11 – Ranking App: Radar Plot ...................................................................... 37
Figure 12 – Ranking App: Two sided bar chart .......................................................... 38
Figure 13 – UML class diagram of arules package .................................................... 45
Figure 14 – Matrix Visualization technique .............................................................. 48
Figure 15 – 3D Matrix Visualization Technique ....................................................... 49
Figure 16 – Graph Visualization technique .............................................................. 50
Figure 17 – Grouped Matrix Visualization Technique ................................................. 51
Figure 18 – Graph plot / rules with high lift ............................................................... 56
Figure 19 - Grouped plot matrix / 10 rules ............................................................... 56
Figure 20 – Grouped matrix plot .............................................................................. 57
1 Introduction

The increasing global competition and the continuous progress of technology, lead to the rapid usage of online training platforms. The purpose of this thesis, is to present useful statistical and machine learning techniques that can be used in adaptive learning environments.

These techniques are going to be used in the Competence Oriented Multilingual Adaptive Language Assessment and Training (COMALAT) system, which is described in Chapter 2. However, the tools are designed in a way that can be used in other online training platforms as well.

One of the problems that these platforms face is the lack of a physical evaluator. Given that flexible evaluating methods are considered necessary in training platforms this is an obstacle we must overcome. In Chapter 3, we describe the implementation of a fuzzy grading system that is going to be used in COMALAT system as an intelligent evaluating system of the platform. First, we present a literature review on fuzzy evaluation methods and adaptability in e-learning environments and afterwards we describe the technique that we used in order to apply it in COMALAT platform. Aside from the COMALAT platform, the fuzzy grading system is an innovative evaluation method that can be deployed in other online learning platforms as well, due to its adaptability and capability to automatically set the difficulty of a quiz, based on the learners’ performances.

In Chapter 4, we present a useful ranking system that is going to be used in COMALAT system, as a tool to motivate the learners of the learning platform.

In Chapter 5, we use association rules to find interesting relationships among the data, that will support the learning procedure. More specifically, firstly we introduce the readers to the association rules theory and we present them the two packages that are used. Then, analytically we describe the method that we follow to implement association rules mining to the data form the e-learning platform.

Finally, in Chapter 6 we present the outcomes that resulted from the extensive research on the aforementioned subject, and the productive collaboration with my supervisor and all the members of the COMALAT team.
The COMALAT Platform

The aim of the COMALAT project is to develop a language training platform, which is able to adapt to the needs of the learner. The learning platform provides language job-oriented skills through customized learning materials that are tailored to the individual interests, strengths and weaknesses of the learner within the learning progress.

2.1 COMALAT Tools

In the COMALAT project we have selected, after a careful evaluation of alternatives, to build the COMALAT system based on the Sakai Learning Management System. Sakai offers an open and extensible platform in which new components can be added in the form of tools. It is open source with a permissive license, which basically allows reusing the Sakai system at no cost for any purpose. In addition, Sakai LMS already provides the necessary tools for a learning platform, such as the Lesson Builder tool to create language lessons that are displayed to the learners and the SAMigo tool which is the tool that can be used for the creation of various types of tests and quizzes with which the learners can be assessed.

The COMALAT project will contribute two such additional tools.

1. COMALAT Guide Tool: This tool will be used to guide the users through the various steps of the learning process. The basic idea behind the Guide tool is that it will try to fill as much as possible the role of the instructor, by presenting to the user appropriate learning materials in accordance to their progress and by providing marks using fuzzy grading techniques.

2. COMALAT Authoring Tool: This tool will be used by learning content providers that will create the learning materials and tag them appropriately so that they can be used for the adaptability of the learning paths according to users’ needs and performance.
In addition to the usual Sakai Profile information (e.g. photo, email etc.) COMALAT learners specify additional data that can be used for job-specific language learning adaptation. For example, in Figure 1 we can see that the user specifies the desired difficulty level (currently Beginner and Intermediate are supported) and also the job-specific preference\(^1\) (currently Health, Tourism and Hospitality, Science and Technology and Business and Professional Language are supported). In addition, the learner specifies the instructions language (i.e. the language in which instructions or explanations are given – currently English, German and Spanish are supported). With this and other information the COMALAT Guide tool adapts the learning material presented by the Lesson tool to the learners, so that it is specific to their needs. For example, if the users have selected the “Science and Technology” job-specific preference, then in the Intermediate level they will get learning materials customized in this specific job preference, in addition to the usual material that all users get regardless their job preference, furthermore, after users complete the various tests and quizzes the results are recorded and fuzzy grading is used to evaluate the current learner’s fuzzy grades. This process is described in detail in the following Chapter 3.

\(^1\) Language for specific preference is a section in every lesson in the intermediate level of the platform devoted to job-oriented materials. Activities, tasks, or materials to learn about purpose-specific vocabulary or discourse variation. Four domains have been included in the system: Language for Health, Language for Tourism and Hospitality, Science and Technology, and Business and Professional Language.
2.2 Role in COMALAT project

Our role in COMALAT project is to develop the “Statistical Inference and Machine Learning” component, which comprises of two sub-components; the “Statistical Analysis Component”, that collects statistical information about individual learners and collective data about all learners, and the “Machine Learning Component”, that provides the “intelligence” in the system with machine learning techniques.

Specifically, the Statistical Inference and Machine Learning Engine (SIC and MLC correspondingly for its components), is a toolbox of data-intensive methods and algorithms providing intelligence to the entire system. The development of the entire engine will be based on the statistical language R.

The SIC contains a set of relatively simple to understand graphical and statistical techniques for the analysis of learners’ data. The main requirement of SIC is its easy accessibility and interpretability by different actors.

The basic features of SIC and MLC are:

- Data are stored in one data matrix for the learners
- The SIC contains functions of the R statistical language for graphical representation of the data and descriptive statistics.
- The output of SIC statistical analysis is different for different actors. Learners can see only output related to their own data and their relative position to other learner’s data (for example their own ranking relatively to other learners’ performance). Instructors can require and see summary statistics for sets of learners, and also for the evaluation of the learning material.
- The MLC contains advanced functions and algorithms of the R statistical language for the analysis of learners’ data matrix. Their role is to discover trends and possible problems of the learning path and to provide recommendations in order to improve the performance of the system. These involve clustering of learners regarding their performance, clustering of lessons regarding their evaluation, discovering causes of failures through correlation analysis, predicting the future performance from preliminary demographic data through classification algorithms, etc.
2.3 COMALAT Structure

In this section we are going to present the structure of the COMALAT platform.

The system supports 3 languages:

- English
- German
- Spanish

Every language has 2 levels:

- Beginner
- Intermediate

Every level has 2 courses.

Every course has 5 lessons and every lesson has 4 sections for Beginner Level and 5 Sections for Intermediate level:

- Grammar & Functions
- Vocabulary
- Reading & Writing
- Listening & Speaking
- Language for specific purposes (Only for Intermediate Level)

Every section has three subsections.

In Figure 2 we can see the language structure and in Figure 3 is described in more detail the section structure. Moreover, in Figure 2 you can also see the decision points. These points, are the levels where the tools (fuzzy grading, ranking, association rules) will be more effective.

In Figure 2, some areas are marked as decision points. Decision points are the places that the Statistical & Machine Learning Techniques can be used with the most effective way.

Furthermore, the learners are evaluated through quizzes that are placed in different places in the structure of the system. Specifically, there are quizzes in the subsections of every lesson and in every lesson, lesson quiz between the lessons, course quiz between the courses, and two final quizzes one for the beginner level and one for the intermediate. The learners, has to pass a quiz in order to continue his/her learning path.
2.4 User Tracking Information in COMALAT

In this section, will be described what kind of information is collected about the learners while they are proceeding through the COMALAT platform. The key pieces information that will be collected for each learner in the system are presented in the Table 1.
Table 1 – Key pieces of information

<table>
<thead>
<tr>
<th>Key pieces of information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demographics</td>
<td>Personal characteristics such as sex, age, education level, current occupation and current place of residence, target occupation, target place of residence</td>
</tr>
<tr>
<td>Current language competencies</td>
<td>Mother tongue, level of competency of other languages</td>
</tr>
<tr>
<td>Target language competencies</td>
<td>Target language, Required level</td>
</tr>
<tr>
<td>Personal learning material</td>
<td>The exact activities of the entire learning path. In this regard, each piece of learning material or activity (tests, etc.) in the system should be characterized by a unique codename. Then, the data for each learner would contain a series of 1s when the learner followed the specific material/activity and 0s otherwise.</td>
</tr>
<tr>
<td>Time spent on activities (hours)</td>
<td>Attempts in lessons are tracked. Time spent on assessment and testing is available in user history information or log files</td>
</tr>
<tr>
<td>Learner’s performance</td>
<td>Marks characterizing the learner’s performance after an assessment procedure. Marks could be represented by a 0-100 percentage scale.</td>
</tr>
<tr>
<td>Learning material evaluation</td>
<td>Each learner provides feedback regarding the learning material, i.e. evaluates with a ranking the difficulty and the appropriateness of the material.</td>
</tr>
</tbody>
</table>
3 Fuzzy Grading

3.1 Introduction to Fuzzy Logic

In order to understand the fuzzy grading system, here, we present an introduction about what fuzzy logic is.

Fuzzy logic is a form of many-valued logic in which the truth values of variables may be any real number between 0 and 1. Contrary to that, in Boolean logic, the truth values of variables may only be 0 or 1. Fuzzy logic is an approach to computing based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean logic on which the modern computer is based.

The idea of fuzzy logic was first introduced by Dr. Zadeh [1] at 1960s. Fuzzy logic includes 0 and 1 as extreme cases of truth, but also includes the various states of truth in between so that, for example, the result of a comparison between two temperatures could be not "cold" or "hot" but ".27 of hotness".

Fuzzy logic seems closer to the way that human brain work. We aggregate data and form a number of partial truths which we aggregate further into higher truths which in turn, when certain thresholds are exceeded, cause certain further results such as motor reaction. A similar kind of process is used in artificial computer neural network and expert systems.

3.1.1 Fuzzy logic system

A fuzzy logic system (FLS) can be defined as the nonlinear mapping of an input data set to a scalar output data. A FLS consists of four main parts:
CHATZISTAVROU KYRIAKI

- fuzzifier
- rules
- inference engine
- defuzzifier

These components and the general architecture of a FLS is shown in Figure 5.

![Figure 5 – Example of a Fuzzy Logic System](image)

The process of fuzzy logic is explained in the algorithm below.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Define the linguistic variables and terms (initialization)</td>
</tr>
<tr>
<td>2.</td>
<td>Construct the membership functions (initialization)</td>
</tr>
<tr>
<td>3.</td>
<td>Construct the rule base (initialization)</td>
</tr>
<tr>
<td>4.</td>
<td>Convert crisp input data to fuzzy values using the membership functions (fuzzification)</td>
</tr>
<tr>
<td>5.</td>
<td>Evaluate the rules in the rule base (inference)</td>
</tr>
<tr>
<td>6.</td>
<td>Combine the results of each rule (inference)</td>
</tr>
<tr>
<td>7.</td>
<td>Convert the output data to non-fuzzy values (defuzzification)</td>
</tr>
</tbody>
</table>

Firstly, a crisp set of input data are gathered and converted to a fuzzy set using fuzzy linguistic variables, fuzzy linguistic terms and membership functions. This step is known as fuzzification. Afterwards, an inference is made based on a set of rules. Lastly, the resulting fuzzy output is mapped to a crisp output using the membership functions, in the defuzzification step. [2]

### 3.1.2 Linguistic Variables

Linguistic variables are the input or output variables of the system whose values are words or sentences from a natural language, instead of numerical values. A linguistic variable is generally decomposed into a set of linguistic terms.

For instance, let temperature (t) (Figure 4) be the linguistic variable which represents the temperature of a room. To qualify the temperature, terms such as “hot” and “cold” are used in real life. These are the linguistic values of the temperature. Then, $T(t) =$
\{cold, warm, hot\} can be the set of decompositions for the linguistic variable temperature. Each member of this decomposition is called a linguistic term and can cover a portion of the overall values of the temperature. [2]

### 3.1.3 Membership Functions

Membership functions are used in the fuzzification and defuzzification steps of a FLS, to map the non-fuzzy input values to fuzzy linguistic terms and vice versa. A membership function is used to quantify a linguistic term. Note that, an important characteristic of fuzzy logic is that a numerical value does not have to be fuzzified using only one membership function. In other words, a value can belong to multiple sets at the same time. There are different forms of membership functions such as triangular, trapezoidal, piecewise linear, Gaussian, or singleton (Figure 6). The most common types of membership functions are triangular, trapezoidal, and Gaussian shapes. [2]

![Figure 6 – Different types of membership functions](image)

### 3.1.4 Fuzzy Rules

In a FLS, a rule base is constructed to control the output variable. A fuzzy rule is a simple IF-THEN rule with a condition and a conclusion. For example, in our example FLS () fuzzy rules could have the form shown in Table 2, which shows the matrix representation of the fuzzy rules for the said FLS. Row captions in the matrix contain the values that current room temperature can take, column captions contain the values for target temperature, and each cell is the resulting command when the input variables take the values in that row and column. For instance, the cell (2,3) in the matrix can be read as follows: “If temperature is cold and target is warm then command is heat.” [2]

<table>
<thead>
<tr>
<th>Temperature/target</th>
<th>Cold</th>
<th>Warm</th>
<th>Hot</th>
</tr>
</thead>
</table>
### 3.1.5 Fuzzy Set Operations

The evaluations of the fuzzy rules and the combination of the results of the individual rules are performed using fuzzy set operations. The operations on fuzzy sets are different than the operations on non-fuzzy sets. Let $\mu_A$ and $\mu_B$ be the membership functions for the fuzzy sets A and be respectively. Table 3 contains possible fuzzy operations for OR and AND operators on these sets, comparatively. The most used operations for OR and AND operators are max and min, respectively. For complement operation, i.e. NOT $\mu_A(x) = 1 - \mu_A(x)$ used for fuzzy sets.

<table>
<thead>
<tr>
<th></th>
<th>OR (Union)</th>
<th>AND (Intersection)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX</td>
<td>min(1, $\mu_A(x) + \mu_B(x)$)</td>
<td>MIN  min($\mu_A(x)$, $\mu_B(x)$)</td>
</tr>
<tr>
<td>ASUM</td>
<td>$\mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x)$</td>
<td>PROD   $\mu_A(x)\mu_B(x)$</td>
</tr>
<tr>
<td>BSUM</td>
<td>min(1, $\mu_A(x) + \mu_B(x)$)</td>
<td>BDIF max(0, $\mu_A(x) + \mu_B(x) - 1$)</td>
</tr>
</tbody>
</table>

After evaluating the result of each rule, these results should be combined to obtain a final result. This process is called inference. The results of individual rules can be combined in different ways. [2]

### 3.1.6 Defuzzification

After the inference step, the overall result is a fuzzy value. This result should be defuzzified to obtain a final crisp output. This is the purpose of the defuzzifier component of a FLS. Defuzzification is performed according to the membership function of the output variable. Note that, there are different algorithms for defuzzification. [2]

### 3.2 Introduction to Fuzzy Grading

In online platforms a fair and innovative method that will automatically evaluate the learners is essential. The main problem that online platforms face is the lack of a human
evaluator, and because of his absence, the decision if a learner with a score close to the baseline - but below it - will pass the course or not, cannot be made. So, for instance, a learner A with a score 49.9 and a learner B with a score 50.1 are actually at the same level. If the platform uses a deterministic model to evaluate the learners, only learner B will pass the course. To avoid that incidence, the platform is going to use a fuzzy model for the learners’ evaluation. The model will be constructed using Law’s [3] method, which is, after an extensive research, the most suitable for the plethora of the training platforms.

3.3 Related Work

Fuzzy set theory was introduced in 1965 by Zadeh [1], and it has been widely used in solving problems in various fields. Recently, fuzzy set theory has been adopted in educational grading systems.


An efficient algorithm of the fuzzy grading system has been suggested by C.K. Law [3]. The study describes the building of a fuzzy structure model for an educational grading system with its algorithm to aggregate different test scores in order to produce a single score for individual students. Law also proposed a method to build the membership functions (MFs) of several linguistic values with different weights. Moreover, Law’s study was introduced not only to assist practical setting by, aggregating different test scores in order to produce a single score for a student, but also to help deciding whether to revise an instructional procedure for the learners.


Chen and Lee [6] presented two methods for applying fuzzy sets to overcome the problem of giving two different fuzzy marks to students with the same total score, which could arise from Biswas’ method.

Wang and Chen [7] presented a method for evaluating students’ answer scripts using fuzzy numbers associated with degrees of confidence of the evaluator. The satisfaction levels awarded to the questions of students’ answer scripts are represented by fuzzy numbers associated with degrees of confidence between zero and one. The arithmetic
operations between the $\alpha$-cuts of fuzzy numbers are used to evaluate the total mark of each student, where $\alpha \in [0,1]$. The proposed method can overcome the drawbacks of the methods presented by Biswas [4] and Chen and Lee [6]. It can evaluate students’ answer scripts in a more flexible and more intelligent manner.

Weon and Kim [8] developed an evaluation strategy based on fuzzy MFs. They pointed out that the system for students’ achievement evaluation should consider the three important factors of the questions given to students: the difficulty, the importance, and the complexity. Weon and Kim used singleton functions to describe the factors of each question reflecting the individual effect of the three factors, but not the collective effect.

Bai and Chen [9] proposed a method to automatically construct the grade membership functions of lenient-type grades, strict-type grades and normal-type grades of fuzzy rules, respectively, for students’ evaluation. Based on the constructed grade membership functions, the system performs fuzzy reasoning to infer the scores of students. It provides a useful way to evaluate students’ answer scripts in a smarter and fairer manner.

Bai and Chen [10] pointed out that the difficulty factor is a very subjective parameter and may cause an argument concerning fairness in evaluation. Moreover, they present a new method for dealing with students’ learning achievement evaluation using fuzzy membership functions and fuzzy rules. The proposed method considers the difficulty, importance and complexity of questions for students’ answer scripts evaluation. It provides a useful way to distinguish the ranking order of students with the same score. However, their method still has the subjectivity problem, since the results in scores and ranks are heavily dependent on the values of several weights, which are determined by the subjective knowledge of domain experts. Specifically, Bai and Chen’s method has seven weights, two for each of the main steps and one for the grade adjustment step, which are determined subjectively by domain experts. Quite different ranks can be obtained depending on their values. By using their method, the examiners could not easily verify how new ranks are obtained and could not persuade skeptical students. Naturally, there is no method to determine the optimum values of weights. Also, the membership values in Bai and Chen’s method do not satisfy the concept of the fuzzy set.

Saleh and Kim [11] as an improved alternative to Bai and Chen’s method [9], proposed a fuzzy logic evaluation system considering the importance, the difficulty, and the complexity of questions, based on Mamdani’s fuzzy inference [12] and center of gravity
(COG) defuzzification. The transparency and objective nature of the fuzzy logic system make it easy to understand and explain the results of evaluation, and thus to persuade students who are skeptical or not satisfied with the evaluation results. The system consists of three nodes: the difficulty node, the effort node, and the adjustment node. Each node of the system behaves like a fuzzy logic controller (FLC) with two scalable inputs and one output. It maps a two-to-one fuzzy relation by inference through a given rule base. The inputs to the system, are given either by examination results or domain experts. The inputs are fuzzified based on the defined levels (fuzzy sets). In the first node, both inputs are given by examination results, whereas in the later nodes, one input is the output of its previous node and the other is given by a domain expert. The output of each node can be in the form of a crisp value (defuzzified) or in the form of linguistic variables (MFs). Each node has two scale factors (SFs), and we can adjust the effects of inputs by varying the values of the scale factors.

Shyi-Ming Chen [13] presented a new method for evaluating students’ learning achievement by automatically generating the weights of the attributes “accuracy rate”, “time rate”, “difficulty”, “complexity”, “answer-cost” and “importance” of fuzzy rules, respectively, with the fuzzy reasoning capability, which provide more fairer and more reasonable results for students’ learning achievement evaluation than Saleh and Kim’s method [11].

In our work we follow Law’s approach to fuzzy grading [3] with the difference that we use the evaluation method in an e-learning system. Moreover, in our study we use Law’s method [11] in order to let the evaluation method be more rigorous or more lenient in a particular quiz, according to other learners’ performance.

3.4 Implementation of Law’s Method in Fuzzy Grading with R Programming Language

To implement fuzzy grading, the system should be provided by:

a) A vector defined by the instructor containing the predetermined percentages of the ideal population receiving grades A, B, C, D, and F.

b) A vector defined by the instructor containing the weights of the tests of the course.

c) A vector containing the learner’s raw grades.
Let us assign to a vector the predetermined percentages $p_A, p_B, p_C, p_D,$ and $p_F$. These percentages represent the ideal mixture of the population receiving the grades A, B, C, D, and F, respectively. The fuzzy grading system then transforms the nominal values A, B, C, D and F into corresponding reasonable normal fuzzy numbers $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$ and $\tilde{F}$ with trapezoidal (or triangular) membership functions $\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x), \mu_{\tilde{C}}(x), \mu_{\tilde{D}}(x)$ and $\mu_{\tilde{F}}(x)$.

In order to find the final scores or grades of learners, the centroid method to defuzzify learners' aggregative scores is employed. Therefore, the expectation values of the normal fuzzy numbers $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$ and $\tilde{F}$ are calculated. Let us denote the expectation of the normal fuzzy numbers $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$ and $\tilde{F}$ by $E(\tilde{A}), E(\tilde{B}), E(\tilde{C}), E(\tilde{D}),$ and $E(\tilde{F})$.

To simplify the notations let $p_A = a$, $p_B = b$, $p_C = c$, $p_D = d,$ and $p_F = f$.

The work of Law in [3] covers the analytic construction of the membership functions and the expectations values, which is used in this case.

Here, as an example, in Listing 1, we present a code snippet that constructs the membership function of A and its expected value, implemented in R.

```r
# iprg: vector with the ideal percentages of receiving grades A,B,C,D,F
# the instructor sets the values of the vector
iprg<-c(a,b,c,d,f)

## membership functions and expected values
## x ε [0,1]
## for a
me_a<-function(x,iprg){
  if(2*a<=b){
    if(x>=0 & x<=(1-(2*a))){
      mf_a<-0
    } else{
      mf_a<-1+((x-1)/(2*a))
    }
    e_a<--(3-2*a)/3
    return (c(mf_a, e_a))
  } else{
    if(x>=0 & x<=1-(b/2)){
      mf_a<-0
    }
    return (c(mf_a, e_a))
  }
}
```

[2] https://www.r-project.org/
else if \((x > (1 - a - (b/2))) \&\& x <= (1 - a + (b/2)))\)
    \(mf_a <= 1 + ((x - (1 - a + (b/2))) / b)\)
else{
    \(mf_a <= 1\)
}
\(e_a <= (24*a - 12*a^2 - b^2) / 24 * a\)
return \(c(mf_a, e_a)\)

Listing 1. Code snippet – constructing the membership functions

Let \(S_1, S_2, \ldots, S_N\) be the raw scores of a learner in a test, and let their corresponding true scores be \(s_1, s_2, \ldots, s_N\). Then, we can form a fuzzy assessment matrix \(M\) as

\[
M = \begin{pmatrix}
    X_1 & \mu_\tilde{A}(s_1) & \mu_\tilde{B}(s_1) & \mu_\tilde{C}(s_1) & \mu_\tilde{D}(s_1) & \mu_\tilde{F}(s_1) \\
    X_2 & \mu_\tilde{A}(s_2) & \mu_\tilde{B}(s_2) & \mu_\tilde{C}(s_2) & \mu_\tilde{D}(s_2) & \mu_\tilde{F}(s_2) \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    X_N & \mu_\tilde{A}(s_N) & \mu_\tilde{B}(s_N) & \mu_\tilde{C}(s_N) & \mu_\tilde{D}(s_N) & \mu_\tilde{F}(s_N)
\end{pmatrix}
\]

As we can see in Listing 2, an \(N \times 5\) matrix \(M\) is created, where \(N\) is the number of tests that a course has.
Listing 2. Code snippet – matrix M construction

For test $X_j$, the fuzzy assessment matrix $M$ indicates the degrees of membership of the learner's corresponding true score $s_j$ in each grades. The student's aggregative score will be defuzzified by employing the centroid method as follows. In Listing 3 you can see the calculation of vector $T$.

Let

$$T = (T_1, T_2, ..., T_N)^t = M \times \left( E(A), E(B), E(C), E(D), E(F) \right)^t$$

Where

$$T_j = \mu_A(s_j)E(A) + \mu_B(s_j)E(B) + \mu_C(s_j)E(C) + \mu_D(s_j)E(D) + \mu_F(s_j)E(F) \text{ for } j = 1, 2, ..., N$$

Listing 3. Code snippet – calculation of vector T

Finally, the learner's aggregative score is $w = \sum_{j=1}^{N} w_j T_j$, where $w_1, w_2, ..., w_N$ be the weightings of the tests. In Listing 4 we present the implementation of the final score calculation.

Listing 4. Code snippet – final score calculation

In section 3.5 and in Table 4 we can see an example showing how a raw grade becomes a fuzzy grade according to the aforementioned method.
3.5 Usage of Fuzzy Grading and Examples

The fuzzy grading system can be used in several cases in training platforms or even as a tool for an instructor. In this section, we describe the cases that fuzzy grading system can be used with the most efficient way.

As we described in the introduction the fuzzy grading provides flexibility in evaluation. Because of the lack of a human evaluator the decision making without the fuzziness is deterministic. For instance, if we have the performances in the same quiz in beginner level from two learners, the learner A with a score 59.95% and the learner B with a score 60.00%, if a deterministic model will be used in the platform only the learner B could pass the quiz, even thou the two learners’ performance are quite the same.

Moreover, the system can determine automatically the difficulty of the quizzes and provide adaptability to the system. To be exact, in learning platforms there are many cases that a learner cannot pass the quiz that is assigned to him/her, not because of his/her lack of knowledge, but because of the complexity or/ and difficulty of the quiz. So, based on other learners’ performance we could use fuzziness to make a quiz quite easier (or quite harder) to pass, without changing a lot the overall learners’ score.

In order to use the fuzzy grading system, in this particular way, when a specific predetermined number of learners will keep failing a quiz, this quiz will be tagged as difficult. Then, we will set certain predetermined vector of percentages receiving the grades A, B, C, D, F (see section 4), and by this way we will make the quiz quite easier for the learners to pass it.

Table 1 shows the raw scores of 10 learners and their corresponding grade according to fuzzy grading system with the characteristics that have described above. The values G1, G2, G3, G4 and G5 represent the learner’s row scores in test 1, test 2, test 3, test 4 and test 5, respectively, of a course. The sum represents the sum of the learner’s raw grades. The w represents the aggregative score, as it has described above. The grade represents the final grade, that is assigned to the learner. The ideal percentages of receiving grades A, B, C, D, F are 15%, 35%, 20%, 20%, 10%, respectively. We also used weights for the tests (test 1: 10%, test 2: 15%, test 3: 20%, test 4: 25%, test 5: 30%).

<table>
<thead>
<tr>
<th>G1</th>
<th>G2</th>
<th>G3</th>
<th>G4</th>
<th>G5</th>
<th>Total raw</th>
<th>w (Aggregative)</th>
<th>Grade</th>
</tr>
</thead>
</table>

3 In that case, A:>80%, B: 50-80%, C: 30-50%, D: 30-10%, F: <10% based on [1].
In Table 5 we can see the usage of fuzzy grading in some extreme cases. If, for example, a learner in Beginner level, in which in order to pass the test he/she should achieve a grade greater than 60%\(^4\), achieves 55.4\% in a “difficult” quiz (as defined in the previous paragraphs) with fuzzy grading takes a 60.99\% and passes the quiz.

<table>
<thead>
<tr>
<th>id</th>
<th>Level</th>
<th>Percentages</th>
<th>Grades</th>
<th>Weights</th>
<th>Real Grade</th>
<th>Fuzzy grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Beginner (&lt;60%)</td>
<td>A: 16%, B: 25%, C: 20%, D: 27%, F: 12%</td>
<td>41 49 67 58 62</td>
<td>no weights</td>
<td>55.4%</td>
<td>60.99%</td>
</tr>
<tr>
<td>2</td>
<td>Beginner (&gt;60%)</td>
<td>A: 16%, B: 25%, C: 20%, D: 27%, F: 12%</td>
<td>41 49 67 58 62</td>
<td>10 15 20 25 30</td>
<td>57.95%</td>
<td>64.56%</td>
</tr>
<tr>
<td>3</td>
<td>Intermediate (&gt;70%)</td>
<td>A: 21%, B: 18%, C: 10%, D: 30%, F: 21%</td>
<td>71 49 67 88 62</td>
<td>no weights</td>
<td>67.4%</td>
<td>70.37%</td>
</tr>
<tr>
<td>4</td>
<td>Intermediate (&gt;70%)</td>
<td>A: 21%, B: 18%, C: 10%, D: 30%, F: 21%</td>
<td>71 49 67 88 62</td>
<td>10 15 20 25 30</td>
<td>68.45%</td>
<td>73.56%</td>
</tr>
</tbody>
</table>

**Table 5 - Using Fuzzy Grading in extreme cases**

In order to present the idea of the fuzzy grading and to provide a fuzzy evaluation tool, we developed a standalone system with the programming language R and Shiny\(^5\) package. This application is hosted at: https://comalat.shinyapps.io/fuzzy_grading/

---

\(^4\) In ideal percentages 60% is B, based on [1].

\(^5\) [http://shiny.rstudio.com/](http://shiny.rstudio.com/)
3.6 Fuzzy Grading Web Application

In order to present the idea of the fuzzy grading system and to provide a fuzzy evaluation tool, we developed a standalone system in R and Shiny package, which implements the idea described in the previous sections of the document.

This application is hosted at:

https://comalat.shinyapps.io/fuzzy_grading/

The code that implements the fuzzy grading web application can be found in Appendix A.

In Figure 2 you can see the environment of the web application.

In this application you can set

a) The percentages of receiving the grades A, B, C, D, and F, using the sliders.

b) The actual grades (in a scale 1-100), using the first text input area to enter the comma separated list of the five grades.

c) The comma separated list of five weights of the actual grades (in scale 1-100); the weights should add up to 100, using the second text input area.

d) Whether you want to have weights, or not, using the checkbox.

Once you have set the parameters, at the output window (at the right side) you can see:

(a) the values that you have already set,

(b) the real grade, and

(c) the fuzzy grade.

---

* http://shiny.rstudio.com/
3.7 Predetermined Percentages of Fuzzy Grading

The predetermined percentages are set by the evaluators before the examination. To be more specific, Law said in his paper “Generally, we assign the linguistic values A, B, C, D, and F to describe a student's performance. It is important that the criteria of the performance of the ideal population (students who take the same course in the same school or district) be set before our students take an examination. Thus, the criteria cannot be influenced by how well the subjects in the sample (students in a particular class) do on the examination.” [3]

The predetermined percentages are set by the evaluators in order to adjust the distribution of the grades. For example, if the evaluators know (according to previous knowledge) that the quiz is difficult and 80% fails the test, but they want most of the learners (e.g.
(80%) to pass the quiz they set the percentages of the ideal population receiving the
grades A, B, C, D, and F, as it follows:

\[ p_A = 34\%, p_B = 34\%, p_C = 12\%, p_D = 10\%, p_F = 10\% \]

However, the fact that the evaluators set the predetermined percentages in a way that a
large number of learners to pass the quiz means that a learner with an excellent grade
(before fuzzy grading) may take the same fuzzy grade with a learner with a good grade
(before fuzzy grading).

In contrast to that, if the quiz is easy, and the evaluators want only a small percentage
of the learners to pass the quiz (e.g. 20%), then the predetermined percentages of the
ideal population receiving the grades A, B, C, D, and F could be:

\[ p_A = 10\%, p_B = 10\%, p_C = 20\%, p_D = 30\%, p_F = 30\% \]

Although, we propose that the fuzzy grading will be triggered only if a predetermined
number of learners will fail a quiz, which will be defined as difficult, because in that
way the fuzzy grading system will have the best performance.

### 3.8 Testing

The fuzzy grading algorithm which described analytically in the previous sections was
tested through artificial simulated dataset in order to investigate the behaviour of the
algorithm.

Regarding the simulation mechanism, the procedure derived pure instances of Real
Grades for a sample of 100 students. In the first dataset, the mechanism produced pure
instances via a uniform distribution according to the predefined percentages described
in Table 6 for each Class Grade.

Next, the algorithm was tested on different case studies (scenarios) modifying the ideal
percentages (inputs) of the algorithm according to the rules of Table 7. The findings of
the analysis can be found in the Section 2.1.1.

The findings of the analysis can be found in great detail in the Appendix B.
**Table 6 – Simulated dataset**

<table>
<thead>
<tr>
<th>Simulated Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of Instances</strong></td>
</tr>
<tr>
<td><strong>Number of Pure Instances for Grade A</strong></td>
</tr>
<tr>
<td><strong>Number of Pure Instances for Grade B</strong></td>
</tr>
<tr>
<td><strong>Number of Pure Instances for Grade C</strong></td>
</tr>
<tr>
<td><strong>Number of Pure Instances for Grade D</strong></td>
</tr>
<tr>
<td><strong>Number of Pure Instances for Grade E</strong></td>
</tr>
<tr>
<td><strong>Number of Lessons</strong></td>
</tr>
</tbody>
</table>
Table 7 – Alternative scenarios for ideal percentages

<table>
<thead>
<tr>
<th>Case Study</th>
<th>(\mu_A)</th>
<th>(\mu_B)</th>
<th>(\mu_C)</th>
<th>(\mu_D)</th>
<th>(\mu_F)</th>
<th>Grading</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(2a \leq b) ((a = 0.15))</td>
<td>(b \geq \max{2a,c}) ((b = 0.35))</td>
<td>(c \leq \min{b,d}) ((c = 0.20))</td>
<td>(d \geq \max{c,2f}) ((d = 0.20))</td>
<td>(2f \leq d) ((f = 0.10))</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(2a &gt; b) ((a = 0.10))</td>
<td>(b \leq \min{2a,c}) ((b = 0.15))</td>
<td>(c \geq \max{b,d}) ((c = 0.40))</td>
<td>(2f &lt; d &lt; c) ((d = 0.30))</td>
<td>(2f \leq d) ((f = 0.05))</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>(2a \leq b) ((a = 0.10))</td>
<td>(b \geq \max{2a,c}) ((b = 0.30))</td>
<td>(d &lt; c &lt; b) ((c = 0.25))</td>
<td>(d \leq \min{c,2f}) ((d = 0.20))</td>
<td>(2f &gt; d) ((f = 0.15))</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>(2a &gt; b) ((a = 0.15))</td>
<td>(c &lt; b &lt; 2a) ((b = 0.20))</td>
<td>(c \leq \min{b,d}) ((c = 0.15))</td>
<td>(c &lt; d &lt; 2f) ((d = 0.30))</td>
<td>(2f &gt; d) ((f = 0.20))</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>(2a &gt; b) ((a = 0.15))</td>
<td>(b \geq \max{2a,c}) ((b = 0.40))</td>
<td>(d &lt; c &lt; b) ((c = 0.25))</td>
<td>(d \leq \min{c,2f}) ((d = 0.10))</td>
<td>(2f &gt; d) ((f = 0.10))</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>(2a &gt; b) ((a = 0.10))</td>
<td>(b \leq \min{2a,c}) ((b = 0.10))</td>
<td>(c \leq \min{b,d}) ((c = 0.10))</td>
<td>(d \leq \min{c,2f}) ((d = 0.10))</td>
<td>(2f &gt; d) ((f = 0.60))</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(2a &gt; b) ((a = 0.40))</td>
<td>(c &lt; b &lt; 2a) ((b = 0.20))</td>
<td>(c \leq \min{b,d}) ((c = 0.15))</td>
<td>(d \leq \min{c,2f}) ((d = 0.15))</td>
<td>(2f &gt; d) ((f = 0.10))</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(2a \leq b) ((a = 0.05))</td>
<td>(2a &lt; b &lt; c) ((b = 0.15))</td>
<td>(c \geq \max{b,d}) ((c = 0.60))</td>
<td>(2f &lt; d &lt; c) ((d = 0.15))</td>
<td>(2f \leq d) ((f = 0.05))</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>(2a \leq b) ((a = 0.05))</td>
<td>(2a &lt; b &lt; c) ((b = 0.20))</td>
<td>(b &lt; c &lt; d) ((c = 0.25))</td>
<td>(c &lt; d &lt; 2f) ((d = 0.30))</td>
<td>(2f &gt; d) ((f = 0.20))</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>(2a \leq b) ((a = 0.10))</td>
<td>(b \geq \max{2a,c}) ((b = 0.40))</td>
<td>(d &lt; c &lt; b) ((c = 0.30))</td>
<td>(2f &lt; d &lt; c) ((d = 0.15))</td>
<td>(2f \leq d) ((f = 0.05))</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>(2a \leq b) ((a = 0.05))</td>
<td>(b \geq \max{2a,c}) ((b = 0.20))</td>
<td>(c \leq \min{b,d}) ((c = 0.15))</td>
<td>(c &lt; d &lt; 2f) ((d = 0.20))</td>
<td>(2f &gt; d) ((f = 0.40))</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>(2a \leq b) ((a = 0.05))</td>
<td>(2a &lt; b &lt; c) ((b = 0.15))</td>
<td>(c \geq \max{b,d}) ((c = 0.60))</td>
<td>(d \leq \min{c,2f}) ((d = 0.05))</td>
<td>(2f &gt; d) ((f = 0.15))</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>(2a \leq b) ((a = 0.05))</td>
<td>(2a &lt; b &lt; c) ((b = 0.20))</td>
<td>(b &lt; c &lt; d) ((c = 0.25))</td>
<td>(d \geq \max{c,2f}) ((d = 0.40))</td>
<td>(2f \leq d) ((f = 0.10))</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>(2a &gt; b) ((a = 0.10))</td>
<td>(b \leq \min{2a,c}) ((b = 0.15))</td>
<td>(c \geq \max{b,d}) ((c = 0.40))</td>
<td>(d \leq \min{c,2f}) ((d = 0.10))</td>
<td>(2f &gt; d) ((f = 0.25))</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>(2a &gt; b) ((a = 0.10))</td>
<td>(b \leq \min{2a,c}) ((b = 0.10))</td>
<td>(b &lt; c &lt; d) ((c = 0.20))</td>
<td>(d \geq \max{c,2f}) ((d = 0.50))</td>
<td>(2f \leq d) ((f = 0.10))</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>(2a &gt; b) ((a = 0.10))</td>
<td>(b \leq \min{2a,c}) ((b = 0.10))</td>
<td>(b &lt; c &lt; d) ((c = 0.15))</td>
<td>(c &lt; d &lt; 2f) ((d = 0.25))</td>
<td>(2f &gt; d) ((f = 0.40))</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>
4 Ranking

An effective way to motivate learners to keep trying to improve their performance is to show them their ranking among the other learner, and how close or far (upper or lower) they are from the learners’ average score. As an example of what the learner can see as an information, we developed a standalone system with R programming language and the packages Shiny and ggplot2; hosted at: https://comalat.shinyapps.io/ranking. The code that implements the ranking application can be found in Appendix C.
In Figure 8 you can see the appearance of the ranking application. Figure 9 represents the input and non-graphical area. First you should set the learner’s code (1-1044). Then you can see

a) Your Grades in Test 1, Test 2, Test 3
b) Your GPA
c) Your rank in Test 1, Test 2, Test 3, and your general rank

Figure 9 – Ranking application input-output (non-graphical) area

From Figure 10 to Figure 12 the applications’ graphical area is presented. Here, the learners can see in three different ways (with a histogram, with a radar plot, and with a two sided bar chart) their performance in comparison with the average grade (and in the radar plot in comparison with the best grade too).
Figure 10 – Ranking App: Histograms

Your grades in comparison with the maximum & the AVG grades

Figure 11 – Ranking App: Radar Plot
Figure 12 – Ranking App: Two sided bar chart
5 Association Rules Mining

5.1 Introduction

Association Rule Mining is one of the popular techniques used in data mining. Association rules are very useful in correlation analysis, they can reveal hidden causal relationships and they can be a significant aid for decision making processes. In educational context, a crucial point that has to be addressed is the assignment of the “right” lessons to the learners depending on various information of the learner. The main concern is the assignment of inappropriate lessons, which do not reflect their overall interest or their needs; this may create serious problems such as poor commitment and underachievement. The addressing of such issues is after all, the essence of the intelligence of a system, i.e. the ability to recognize trends and problems from the data that are collected from learners. These data involve individual information (demographics) on the one hand and scores from the learners' performances on the other.

The input of association rules is individual information (demographics) like mother tongue, age, education, occupation etc. and scores from the learners' performances. The output of association rules are logical expressions of the "if-then" form and are very useful in uncovering the systematic effects of learners’ demographics (e.g. mother tongue, age, education, occupation etc.) on their performance on specific lessons. The correct interpretation of the extracted association rules leads to efficient decisions that improve not only the learners but also the entire educational environment, which is another aspect of intelligence. The ultimate goal of such a system should be the recommendation of the most suitable lessons for learners and the prediction of the effort that is needed in order to reach a specific level.

Association rule mining has many advantages as it has been applied to e-learning systems for traditionally association analysis (finding correlations between items in a dataset), including:

a) building recommender agents for on-line learning activities

b) automatically guiding the learner’s activities and intelligently generate and recommend learning materials
c) identifying attributes characterizing patterns of performance disparity between various groups of students

d) discovering interesting relationships from student’s usage information in order to provide feedback to course author

e) finding out the relationships between each pattern of learner’s behavior

f) finding students’ mistakes that are often occurring together

g) optimizing the content of an e-learning portal by determining the content of most interest to the user

From the other hand, association rule mining comes also with some disadvantages:

a) Association rule mining may obtaining non interesting rules or/and huge number of discovered rules.

b) Association Rules mining needs a large dataset in order to discover interesting relationships that will be used in the course recommendation system.

c) The algorithmic performance is one of the main problems in Association Rule mining area, but with the right mining algorithm this problem can be minimized.

For the implementation of Association Rules there are specific packages in R programming language like the R-extension package arules\(^8\) for association rule mining, which provides the infrastructure needed to create and manipulate input datasets for the mining algorithms and resulting itemsets and rules. Furthermore, the R-extension package arulesViz\(^9\), will be also used for association rules visualization.

### 5.2 Theoretical Background

Mining frequent itemsets and association rules is a popular and well researched method for discovering interesting relations between variables in large databases. Piatetsky-Shapiro \([14]\) describes analyzing and presenting strong rules discovered in databases using different measures of interest. Based on the concept of strong rules, Agrawal, Imielinski, and Swami \([15]\) introduced the problem of mining association rules from transaction data as follows.

---

\(^8\) [https://cran.r-project.org/web/packages/arules/index.html](https://cran.r-project.org/web/packages/arules/index.html)

\(^9\) [https://cran.r-project.org/web/packages/arulesViz/index.html](https://cran.r-project.org/web/packages/arulesViz/index.html)
• Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of $n$ binary attributes called items.

• Let $D = \{t_1, t_2, \ldots, t_m\}$ be a set of transactions called the database. Each transaction in $D$ contains a subset of the items in $I$.

• The rule takes the form $X \rightarrow Y$ where $X, Y \subseteq I$ and $X \cap Y = \emptyset$, meaning that $X, Y$ are the itemsets. The sets of items (for short itemsets) $X$ and $Y$ are called antecedent (left-hand-side or LHS) and consequent (right-hand-side or RHS) of the rule.

  \[
  \begin{array}{c|c}
  \text{LHS} & \text{RHS} \\
  \hline
  X & Y \\
  \end{array}
  \]

For example, to illustrate the concepts, we use a small example from the supermarket domain.

• The itemset is $I = \{\text{milk, butter, bread, beer}\}$.

• The database of the aforementioned itemset is shown in Table 8. Table 9 also presents the same transactions with the Table 8 using binarization.

  Table 8 – An example supermarket database with 5 transactions

<table>
<thead>
<tr>
<th>transaction ID</th>
<th>items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>milk, bread</td>
</tr>
<tr>
<td>2</td>
<td>bread, butter</td>
</tr>
<tr>
<td>3</td>
<td>beer</td>
</tr>
<tr>
<td>4</td>
<td>milk, bread, butter</td>
</tr>
</tbody>
</table>

  Table 9 – An example of a collection of supermarket data presented as a binary incidence matrix

<table>
<thead>
<tr>
<th>transactions</th>
<th>milk</th>
<th>bread</th>
<th>butter</th>
<th>beer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
An example rule for the supermarket could be \{milk, butter\} → \{bread\} meaning that if milk and butter is bought, customers also buy bread.

To select interesting rules from the set of all possible rules, constraints on various measures of significance and interest can be used. The best-known constraints are minimum thresholds on support and confidence.

- **Support** is defined on an itemset \(Z\) as the proportion of transactions in the data set which contain the itemset.

\[
supp(Z) = \frac{n_Z}{n}
\]

In the example that is shown in Table 8 the itemset \{milk, bread\} has a support of \(\frac{2}{4} = 0.5\) since it occurs in 40% of all transactions (2 out of 5 transactions).

All itemsets which have a support above a set minimum support threshold are called frequent itemsets. Finding frequent itemsets can be seen as a simplification of the unsupervised learning problem called “mode finding” or “bump hunting” [16]. For these problems each item is seen as a variable. The goal is to find prototype values so that the probability density evaluated at these values is sufficiently large.

- **Confidence** is defined on rules as \(conf(X \rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}\). This can be interpreted as an estimate of the probability \(P(Y \mid X)\), the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS [17].

For example, the rule \{milk, bread\} → \{butter\} has a confidence

\[
\frac{supp(\{milk, bread\} \cup \{butter\})}{supp(\{milk, bread\})} = \frac{\frac{1}{4}}{\frac{1}{4}} = \frac{0.25}{0.5} = 0.5
\]

in the database shown in Table 8, which means that for 50% of the transactions containing milk and bread, the rule is correct.

Each association rule \(X \rightarrow Y\) has to satisfy both constraints, minimum support and minimum confidence, at the same time, and it is called the support-confidence framework.

\[
supp(X \cup Y) \geq \sigma
\]

\[
conf(X \rightarrow Y) \geq \gamma
\]
• At medium to low support values, often a great number of frequent itemsets are found in a database. However, since the definition of support enforces that all subsets of a frequent itemset have to be also frequent, it is sufficient to only mine all maximal frequent itemsets, defined as frequent itemsets which are not proper subsets of any other frequent itemset [18].

• Another approach to reduce the number of mined itemsets is to only mine frequent closed itemsets. An itemset is closed if no proper superset of the itemset is contained in each transaction in which the itemset is contained [19] [18]. Frequent closed itemsets are a superset of the maximal frequent itemsets. Their advantage over maximal frequent itemsets is that in addition to be able to infer all frequent itemsets, they also preserve the support information for all frequent itemsets which can be important for computing additional interest measures after the mining process is finished (e.g., lift [20], or all-confidence [21]).

• A practical solution to the problem of finding too many association rules satisfying the support and confidence constrains is to further filter or rank found rules using additional interest measures. The measure lift [20] is defined as

\[
lift(X \rightarrow Y) = \frac{\text{conf}(X \rightarrow Y)}{\text{supp}(Y)} = \frac{\text{supp}(X \cup Y)}{\text{supp}(X) \cdot \text{supp}(Y)}
\]

and can be interpreted as an estimate for \( \frac{P(E_X \cap E_Y)}{P(E_X) \cdot P(E_Y)} \), i.e. the deviation of the support of the whole rule from the support expected under independence given the supports of the LHS and the RHS. Greater lift values indicate stronger association rules. And to be more specific, in marketing values of lift are interpreted as

- \( lift(X \rightarrow Y) = 1 \): \( X \) and \( Y \) are independent
- \( lift(X \rightarrow Y) > 1 \): complementary effects between \( X \) and \( Y \)
- \( lift(X \rightarrow Y) < 1 \): substitution effects between \( X \) and \( Y \)

### 5.3 Packages

In the following sections, the R-extension packages that will be used for the implementation of Association Rules, arules for association rule mining (see 5.3.1), and arulesViz, for association rules visualization (see 5.3.2), will be presented.
5.3.1 arules
The core package arules provides an object-oriented framework to represent transaction databases and patterns. To facilitate extensibility, patterns are implemented as an abstract superclass associations and then concrete subclasses implement individual types of patterns. In arules the associations itemsets and rules are provided. Databases and associations both use a sparse matrix representation for efficient storage and basic operations like sorting, subsetting and matching are supported. Different aspects of arules were discussed in previous publications [22].

5.3.1.1 arules Infrastructure
With arules package the basic infrastructure is provided, which allows the association mining and the manipulation of the results. The main features of the arules package are:

- **The sparse matrix representation** (from package Matrix by Bates & Maechler [23]) for transactions and associations.
- **Simple and intuitive interface** to manipulate and analyze data, sets of itemsets and rules with subset selection sorting.
- **Interface to two fast mining algorithms.** In arules package there is an interface of free implementations Apriori and Eclat maximal frequent itemsetsby Christian Borgelt [24] [25]. The code is called directly from R by the functions apriori() and eclact() and the data objects are directly passed from R to the C code and back without writing any external files. The implementation can mine frequent itemsets, closed and maximal frequent itemsets (see 5.2). In addition, apriori() can also mine association rules.
- **Flexibility** in terms of adding new quality measures, and additional item and transaction descriptions which can be used for selecting and analyzing resulting associations.
- **Extensible data** structure to allow for easy implementation of new types of associations and interfacing new algorithms [26]
- Provides **comprehensive analysis** and manipulation capabilities for transactions and associations (subsetting, sampling, visual inspection, etc.).
- The package arulesViz, which can be used for visualizations and it will be further developed in section 5.3.2.
In Figure 13 we present a UML class diagram of the arules package [27].

![UML class diagram of arules package](image)

Figure 13 – UML class diagram of arules package

### 5.3.1.2 Representing collections of itemsets in arules

As it was presented in section 5.2 (Table 9), the collection of the itemsets is presented in a binary incidence matrix, which will in general be very sparse with many items and a very large number of rows. A natural number for such data is a sparse matrix format.

For the implementation, the `ngCMatrix` class is chosen, and it is defined in package `Matrix`. The `ngCMatrix` is a compressed, sparse, logical, column-oriented matrix which contains the indices of the TRUE rows and the pointers to the initial indices of elements in each column of the matrix. Despite the column orientation of the `ngCMatrix`, it is more convenient to work with incidence matrices which are row-oriented. This makes the most important manipulation, selecting a subset of transactions from data set for mining, more comfortable and efficient. Therefore, we implemented the class `itemMatrix` providing a row-oriented facade to the `ngCMatrix` which stores a transposed incidence matrix.

For instance, the itemsets in Table 9 will be represented with

- a vector of indices of the non-zero elements (row-wise starting with the first row): \(1,2,2,3,4,1,2,3\)
- the pointers: \(1,3,5,6\), where each row starts in the index vector. The first two pointers indicate that the first row starts with element one in the index vector and ends with element 2 (since with element 3 already the next row starts).

The two aforementioned vectors are stored in `ngCMatrix`. Note that indices for the `ngCMatrix` start with zero rather than one and thus actually the vectors \(0,1,1,2,3,0,1,2\) and \(0,2,4,5\) are stored.
In addition to the sparse matrix, itemMatrix stores labels (e.g. names of items) and handles the necessary mapping between the item label and the corresponding column number in the incidence matrix. Optionally, itemMatrix can also store additional information on items. For example, the category hierarchy in a supermarket setting can be stored, which enables the analyst to select only transactions that contain items from a certain category. Since itemMatrix is used to represent sets or collections of itemsets additional functionality is provided. [28]

- length(): can be used to get the number of itemsets in an itemMatrix
- duplicated(): can be used to find the identical itemsets
- unique(): can be used to remove the duplications
- match(): can be used to find matching elements in two collections of itemsets
- c(): can be used to combine objects by successively appending the rows of the objects, i.e. creating a collection of itemsets which contains the itemsets of all itemMatrix objects.
- recode(): can be used to make two objects compatible by reordering and inserting columns, if two objects contain the same items, but the order in matrix is different or one object is missing items
- size(): can be used to get the actual number of itemsets stored in the itemMatrix
- itemFrequentcy(): calculates the frequency for each item in an itemMatrix
- itemFrecuencyPlot(): produces a bar plot of item count frequencies or support.
- decode(): can be used to subsecently decode column numbers to item labels
- image(): can be used to produce a level plot of an itemMatrix which is useful for quick visual inspection

5.3.1.3 Associations: itemsets and sets of rules

The result of mining transaction data in arules are associations. Conceptually, associations are sets of objects describing the relationship between some items, which have assigned values for different measures of quality. Such measures can be measures of significance (e.g. support), or measures of interestingness (e.g. confidence, lift), or other measures (e.g. revenue covered by the association).

All types of association have a common functionality in arules comprising the following methods [28]:

- summary(): can be used to give a short overview of the set
- inspect(): can be used to display individual associations
- length(): can be used to get the number of elements in the set
- items(): can be used to get each association a set of items involved in the association (e.g. the union of the items in the LHS and the RHS for each rule)
- sort(): can be used to sort the set using the values of different quality measures
- subset() or []: can be used to extract a subset
- union(): is a set operation that is used to get the union of the sets
- intersect(): is a set operation that is used to get the intersection of the sets
- match(): can be used to match elements from two sets
- write(): can be used to write associations to disk in human readable form. To save and load associations in compact form the methods save() and load() from the base package can be used
- write.pmml() and read.pmml() can be used to write and read associations using PMML (Predictive Model Markup Language) via package pmml. [29]

5.3.2 arulesViz
The package arulesViz extends the package arules, which provides various visualization techniques for association rules and itemsets. The package also includes several interactive visualizations for rule exploration. [30]

In the sections that follow, we will use for the visualization techniques the data “groceries” and the apriori algorithm to generate the association rules. The aforementioned, are implemented with the script below.

```r
#install package arulesViz
install.packages("arulesViz")

#load package arulesViz
library("arulesViz")

#load data for the example
data("Groceries")

#run apriori algorithm to generate association rules
rules <- apriori(Groceries, parameter=list(support=0.001,confidence=0.5))
```
5.3.2.1 Matrix-based visualization

Matrix-based visualization techniques organize the antecedent and consequent itemsets on the x and y-axes, respectively. A selected interest measure is displayed at the intersection of the antecedent and consequent of a given rule. If no rule is available for an antecedent/consequent combination the intersection area is left blank.

The visualized matrix is constructed as follows.

1. Create \( \mathcal{R} = \{ (X_1, Y_1, \theta_1), \ldots, (X_2, Y_2, \theta_2), \ldots, (X_n, Y_n, \theta_n) \} \) where \( X_i \) is the antecedent, \( Y_i \) is the consequent and \( \theta_i \) is the selected interest measure for the \( i \)-th rule, \( i = 1, \ldots, n \).
2. Create a \( A \times C \) matrix\(^10\) \( M = (m_{ac}) \), \( a = 1, \ldots, A \) and \( c = 1, \ldots, C \), with one column for each unique antecedent and one for each unique consequent in \( \mathcal{R} \).
3. Populate \( M \) with \( m_{ac} = \theta_i \) where \( i = 1, \ldots, n \) is the rule index and \( a, c \) correspond to the position \( X_i \) and \( Y_i \) in the matrix. \([31]\)

The code below implements the aforementioned visualization technique and the result that generates is shown in the Figure 14.

```r
plot(rules, method = "matrix", measure = "lift")
```

![Matrix Visualization technique](image)

**Figure 14 – Matrix Visualization technique**

\(^{10}\) \( M \) will be a sparse matrix
Also, the code below implements the 3D matrix visualization technique and the result that generates is shown in the Figure 15.

\[
\text{plot(rules, method = "matrix3D", measure = "lift")}
\]

![Matrix with 5668 rules](image)

Figure 15 – 3D Matrix Visualization Technique

### 5.3.2.2 Graph-based visualizations

Graph-based visualization is particularly well suited when the analyst is interested in an aggregated perspective on the most important rules. Graph-based techniques visualize association rules using vertices and edges, where vertices typically represent items or itemsets and edges indicate relationships in terms of rules. Interest measures are typically added to the plot as labels on the edges or by color or width of the arrows displaying the edges. [31]

In the context of association rule mining, graph-based visualization techniques offer a very clear representation of rules for relatively small sets of most important rules, which can be easily selected based on their corresponding lift scores. Figure 16 presents the graph visualization for the most important extracted association rules. In the network graph, itemsets are represented as vertices, whereas rules are represented as directed edges between itemsets. For illustration purposes, we select the 10 rules with the highest lift scores.
The code below implements the aforementioned visualization technique and the result that generates is shown in the Figure 16.

```r
# graphs only work with very few rules
subrules2 <- sample(rules, 10)
plot(subrules2, method="graph")
plot(subrules2, method="graph", control=list(type="items"))
```

![Graph for 10 rules](image)

**Figure 16 – Graph Visualization technique**

### 5.3.2.3 Grouped matrix-based visualization

Traditional matrix-based visualization is limited in the number of rules it can visualize effectively, since large sets of rules typically also have large sets of unique antecedents and consequents. Therefore, in this section we introduce new visualization techniques that enhance matrix-based visualization by grouping rules through k-means clustering, in order to handle large sets of rules. Groups of rules are presented by aggregating rows and columns of the matrix. The groups are nested and organized hierarchically allowing the analyst to explore them interactively by zooming into groups.

To visualize the grouped matrix, we use a balloon plot with antecedent groups as columns and consequents as rows Figure 17. The color of each balloon represents the aggregated interest measure in the group and the size of the balloon shows the aggregated support. Aggregation in groups can be achieved by several aggregation functions (e.g., maximum, minimum, average, median). The number of rules and the most important (frequent) items in the group are displayed as the labels for the columns followed by the number of other
items in that antecedent group. Furthermore, the columns and rows in the plot are reordered such that the aggregated interest measure is decreasing from top down and from left to right, directing the user to the most interesting group in the top left corner. [31] The code below implements the aforementioned visualization technique and the result that generates is shown in the Figure 17. For illustration purposes, we select the 20 rules with the highest lift scores.

```r
## grouped matrix plot
plot(rules, method="grouped")
plot(rules, method="grouped", control=list(k=30))
```

![Grouped Matrix Visualization Technique](image)

Figure 17 – Grouped Matrix Visualization Technique
5.4 The method

In this section, will presented the method to mine interesting rules from e-learning data, and we will focus on how we are going to implement association rules mining in the COMALAT platform (see Chapter 2). Although, with the appropriate configurations the association rules mining method we describe in this section can be used in other online training platforms as well.

Some of the information that presented in the Section 2.4, will be used in the association rule mining. In further work, a model with all the key user information and more attributes can be constructed.

First, a summary table (Table 10) has been created, which integrates the most important information about the activities and the final marks obtained by students in the courses. Notice that we have transformed all the continuous attributes into discrete attributes that can be treated as categorical attributes. Discretization allows the numerical data to be divided into categorical classes that are easier for the instructor to understand.

Table 10 – Attributes used for each learner

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>course</td>
<td>The course that the learner enrolled</td>
<td>English (E)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Spanish (S)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>German (G)</td>
</tr>
<tr>
<td>jsp</td>
<td>The learner’s job of specific purpose</td>
<td>Health (H)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tourism &amp; Hospitality (T)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Science &amp; Technology (S)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Business &amp; Professional Language (B)</td>
</tr>
<tr>
<td>lvl</td>
<td>The level that the learner has achieved so far</td>
<td>Beginner (1)</td>
</tr>
<tr>
<td>grade</td>
<td>The learners overall grade so far</td>
<td>Real number ∈ (0-1)</td>
</tr>
<tr>
<td>grade_g</td>
<td>The learners overall grade so far in Grammar</td>
<td>Real number ∈ (0-1)</td>
</tr>
<tr>
<td>grade_v</td>
<td>The learners overall grade so far in Vocabulary</td>
<td>Real number ∈ (0-1)</td>
</tr>
<tr>
<td>grade_rw</td>
<td>The learners overall grade</td>
<td>Real number ∈ (0-1)</td>
</tr>
</tbody>
</table>
The critical part of the method is the construction of the sparse matrix. The application of Association Rules requires a large amount of data, which means a long-term collection of information from participating learners. At the time of authoring this thesis, while the platform is under development, such data are not available. Thus, we use synthetic data to construct the sparse matrix. A .csv file is presented in the Appendix D (which has the attributes described in Table 10 to test the function), alongside with the code that implements the association rules function.

The input of the association rules function, is a matrix with the form of the .csv file that is placed in the appendix. Firstly, the function converts all the numerical values of the input matrix to categorical.

Then, a new sparse matrix is created by creating for each row and for each attribute a new vector that has so many elements as the possible values of the attribute and the value 0 when the statements is false, and 1 if the statement is true. For instance, let’s say that we have as an input matrix the Table 11, which as you can see has 2 rows (i.e. it contains 2 learner) and 3 columns (i.e. it has 3 attributes). To convert the Table 11 to a sparse matrix (see Table 12) we create for the first row 3 new vectors (one for each attribute):

- vecCOURSE = [1 0 0]
- vecJSP = [1 0 0 0]
- vecLVL = [1 0]

and for the second row we create again 3 vectors:

- vecCOURSE = [0 1 0]
- vecJSP = [0 1 0 0]
- vecLVL = [1 0]

\[ >90\% \rightarrow \text{HP}, \ 70-90\% \rightarrow \text{P}, \ 60-70\% \rightarrow \text{LP} \] (Note that a learner to pass a quiz in the COMALAT should achieve at least 60%)
Following, for each row we bind (`rbind`) the vectors we created as a row for the new sparse matrix:

- Row 1 of sparse matrix: \([1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0]\)
- Row 2 of sparse matrix: \([0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]\)

Table 11 – Example input matrix

<table>
<thead>
<tr>
<th>Language</th>
<th>JSP</th>
<th>LVL</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>H</td>
<td>1</td>
</tr>
<tr>
<td>S</td>
<td>T</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 12 – Example sparse matrix

<table>
<thead>
<tr>
<th>E</th>
<th>S</th>
<th>G</th>
<th>H</th>
<th>T</th>
<th>S</th>
<th>B</th>
<th>L1</th>
<th>L2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

The way of assigning the correct values to the attribute vectors is described in the following code snippets and analytically in the Appendix D in the section Association Rules Mining Code.

```r
#Course
if(j==1){
  #English
  if(data[i,j]=="E"){
    vecCOURSE<-c(1,0,0)
  }
  #Spanish
  else if (data[i,j]=="S"){
    vecCOURSE<-c(0,1,0)
  }
  #German
  else{
    vecCOURSE<-c(0,0,1)
  }
}
#Job of specific purpose
if(j==2){
  #Health
  if (data[i,j]=="H"){
    vecJSP<-c(1,0,0,0)
  }
  #Tourism and Hospitality
  else if (data[i,j]=="T"){
    vecJSP<-c(0,1,0,0)
  }
  #Science and Technology
  else if (data[i,j]=="S"){
    vecJSP<-c(0,0,1,0)
  }
  #Business and Professional Language
  else{
  }
}
```
vecJSP<-c(0,0,0,1)
}

# level
if(j==3){
  # Level 1
  if(data[i,j]==1){
    vecLVL<-c(1,0)
  }
  # Level 2
  else{
    vecLVL<-c(0,1)
  }
}

After the construction of the sparse matrix we convert the matrix into transactions and we call the apriori algorithm for association rules mining from the package arules. At this point, rules are generated and returned as in output of the function.

trans <- as(mat_n[,,-1]>0, "transactions")
rules <- apriori(trans, parameter = list(supp = 0.01, conf = 0.8))

Moreover, the function also returns the rules with the highest lift, two plots one by using the graph visualization technique, and one by using the grouped matrix visualization technique\(^\text{12}\). In order to understand better the outcome of the association rules go to the Appendix D – Association Rules Mining in the section Glossary.

<table>
<thead>
<tr>
<th>lhs</th>
<th>rhs</th>
<th>support</th>
<th>confidence</th>
<th>lift</th>
</tr>
</thead>
<tbody>
<tr>
<td>{H,G_G_LP}</td>
<td>{G_SL_LP}</td>
<td>0.01</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>{H,G_SL_LP}</td>
<td>{G_G_LP}</td>
<td>0.01</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>{E,G_G_LP}</td>
<td>{G_SL_LP}</td>
<td>0.01</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>{E,G_SL_LP}</td>
<td>{G_G_LP}</td>
<td>0.01</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>{TH,G_LP}</td>
<td>{G_G_LP}</td>
<td>0.01</td>
<td>1</td>
<td>50</td>
</tr>
</tbody>
</table>

\(^{12}\) By using the package arulesViz.
Figure 18 – Graph plot / rules with high lift

Figure 19 - Grouped plot matrix / 10 rules
Figure 20 – Grouped matrix plot
6 Conclusion

In this thesis, we presented statistical & machine learning techniques that can be used in Adaptive Learning Environments. Specifically, these techniques will be used in the COMALAT platform, a Competence Oriented Multilingual Adaptive Language Assessment and Training platform, to support the learning procedure and to provide intelligence to the system.

However, the tools are designed in a way that can be used in other online training platforms as well. So, all the tools presented in this thesis can be developed further, be customized, and be used as standalone tools for supporting a physical evaluator or/and in other educational platforms to support the learning procedure.

Due to the continuous evolution of technology and the growth of multinational companies, there is a plethora of training platforms, which are used by the employees or the future employees in order to increase their personal and professional competencies. Nevertheless, these platforms do not use statistical and machine learning techniques to provide an adaptable environment to their learners, in contrary to COMALAT platform, whose aim is to create a learning path for each learner based to his/hers personal strong and weak points.

Fuzzy grading is a very innovative grading system, because it eliminates the lack of the physical evaluator (in the grading procedure) that all the platforms face. Thus, in future work the fuzzy grading system described in this thesis, could be developed further by adjusting, for instance, the membership functions. Moreover, a learning platform can be more adaptable to their learners by auto-adjusting the difficulty of their tests and quizzes using fuzzy grading and/or other techniques.

In Chapter 4, we presented an effective way to motivate learners to keep try and to improve their performance by showing them their ranking among the other learners. We selected some visualization techniques to visualize their rank, but in future work, more visualization techniques could be used to visualize the learners’ rank.

In Chapter 5, we presented how association rule mining could support the learning procedure, and we suggested a certain method of mine the association rules from the data
that will be provide from the platform. The main problem that we faced is the lack of data because the application of Association Rules requires a large amount of data, which means a long-term collection of information from participating learners. At the time of authoring this thesis, while the platform is under development, such data are not available. So, when we collect a large amount of data we could validate the results, use other mining algorithms, or experiment with the parameters of association mining technique. Furthermore, at this point authoring this thesis, we only use a specific number of learner’s attributes. In future work, more attributes could be used. To be more specific, demographic data, such as learners’ age or learners’ mother tongue, may help the discovery of interesting rules, which would help the learning procedure and would provide adaptability to the training platform.
Bibliography


Appendix A – Fuzzy Grading Application

ui.R

```r
library(shiny)
shinyUI(pageWithSidebar(
  headerPanel("Fuzzy Grading"),
  sidebarPanel(
    h5(strong("Insert the ideal percentages of receiving grades A, B, C, D, F:")),
    sliderInput("perc_a", h6(strong("Grade A")), 15, min=0, max=100, step=1, width = '100%'),
    sliderInput("perc_b", h6(strong("Grade B")), 35, min=0, max=100, step=1),
    sliderInput("perc_c", h6(strong("Grade C")), 20, min=0, max=100, step=1),
    sliderInput("perc_d", h6(strong("Grade D")), 20, min=0, max=100, step=1),
    sliderInput("perc_f", h6(strong("Grade F")), 10, min=0, max=100, step=1),
    textInput("grades", "Insert the grades in a 100 scale for a learner separated with commas", "80,80,75,96,90"),
    textInput("weights", "Insert the weights of the grades (The weights should sum up to 100)!", "10,15,20,25,30"),
    checkboxInput("eqWeights", "equal grades - no weights", value = FALSE, width = NULL)
  ),
  mainPanel(
    h3('You entered'),
    h4(textOutput("inputPercentages")),
    h4("Grades: ", textOutput("inputGrades")),
    h4("Weights: ", textOutput("inputWeights")),
    # ayta prepeli na typonontai mono an ta vazh einai swsta!!
    h4("Real Grade: ", textOutput("realGrade")),
    h4("Fuzzy Grade: ", textOutput("inputResults2"))
  )
))
```

server.R

```r
library(shiny)
#interactive()==TRUE

delta<-0.0001

## membership functions and expected values
## x E [0,1]
```
mf_a<-function(x,iprg){
a<-iprg[1]
b<-iprg[2]
c<-iprg[3]
d<-iprg[4]
f<-iprg[5]
if((2*a)<=b){
  if(x>=0 & x<=(1-(2*a))){
    mf_a<-0
  } else{
    mf_a<-1+((x-1)/((2*a)+delta))
  }
  return (mf_a)
} else{
  if(x>=0 & x<=1-(a-(b/2))){
    mf_a<-0
  } else if (x>(1-a-(b/2)) & x<=(1-a+(b/2))){
    mf_a<-1+((x-(1-a+(b/2)))/(b+delta))
  } else{
    mf_a<-1
  }
  return (mf_a)
}
}
ev_a<-function(iprg){
a<-iprg[1]
b<-iprg[2]
c<-iprg[3]
d<-iprg[4]
f<-iprg[5]
if((2*a)<=b){
  ev_a<-(3-2*a)/3
  return (ev_a)
} else{
  ev_a<-(24*a-12*a^2-b^2)/((24*a)+delta)
  return (ev_a)
}
}
## for b
mf_b<-function(x,iprg){
a<-iprg[1]
b<-iprg[2]
c<-iprg[3]
d<-iprg[4]
f<-iprg[5]
if(b>=max((2*a),c)){
  if(x>=0 & x<=(f+d+(c/2))){
    mf_b<-0
  } else if (x>(f+d+(c/2)) & x<=(f+d+(3*c/2))){
    mf_b<-1+((x-(f+d+(3*c/2)))/(c+delta))
  } else if (x>(f+d+(3*c/2)) & x<=(-2*a)){
    mf_b<-1
  } else{
    mf_b<-1-(x-(1-2*a))/(2*(a+delta))
  }
  return (mf_b)
} else if((2*a)<b & b<c){
  if(x>=0 & x<=(1-a-(3*b/2))){
mf_b<=0
} else if (x>(1-a-(3*b/2)) & x<=(1-a-(b/2))){
    mf_b<=1+((x-(1-a-(b/2)))/(b+delta))
} else if(x>(1-a-(b/2)) & x<=((1-(2*a))){
    mf_b<=1
} else{
    mf_b<=1-
((x-(1-(2*a)))/(2*a)+delta))
} return (mf_b)

} else if(c<b & b<(2*a)){
    if(x>=0 & x<=(f+d+(c/2))){
        mf_b<=0
    } else if (x>(f+d+(c/2)) & x<=(f+d+(3*c/2))){
        mf_b<=1+(x-(f+d+(3*c/2)))/(c+delta)
    } else if (x>(f+d+3*(c/2)) & x<=(f+d+c+(b/2))){
        mf_b<=1
    } else if(x>(f+d+c+(b/2)) & x<=(f+d+c+(3*b/2))){
        mf_b<=1-
((x-(f+d+c+(b/2)))/(b+delta))
    } else{
        mf_b<=0
    }

    return (mf_b)
} else{ if(x>=0 & x<=(1-a-(3*b/2))){
        mf_b<=0
    } else if (x>(1-a-(3*b/2)) & x<=((1-a-(b/2))){
        mf_b<=1+((x-(1-a-(b/2)))/(b+delta))
    } else if(x>(1-a-(b/2)) & x<=((1-a-(b/2))){
        mf_b<=1-
((x-(1-a-(b/2)))/(b+delta))
    } else{
        mf_b<=0
    }

    return (mf_b)
}

ev_b<function(iprg){
a<-iprg[1]
b<-iprg[2]
c<-iprg[3]
d<-iprg[4]
f<-iprg[5]
if(b>max((2*a),c)){
    ev_b<=(4*a^2-24*a*b-12*b^2+24*b+c^2)/((24*b)+delta)
    return (ev_b)
} else if((2*a)<b & b<c){
    ev_b<=(4*a^2-24*a*b-13*b^2+24*b)/((24*b)+delta)
    return (ev_b)
} else if(c<b & b<(2*a)){

ev_b = (13*b^2 + 24*b*c + 24*b*d + 24*b*f - c^2) / (24*b + delta)
return (ev_b)
} else
  ev_b = (2 - 2*a - b) / 2
return (ev_b)
}

## for c
mf_c =< function(x, iprg){
a = iprg[1]
b = iprg[2]
c = iprg[3]
d = iprg[4]
f = iprg[5]
if(c >= max(b, d)){
  if(x >= 0 & x <= (f + (d/2))){
    mf_c = 1
  }
  else if (x > (f + (d/2)) & x <= (f + (3*d/2))){
    mf_c = 1 + ((x - (f + (3*d/2))) / (d + delta))
  }
  else if(x > (f + (3*d/2)) & x <= (f + d + c - (b/2))){
    mf_c = 1
  }
  else if(x > (f + d + c - (b/2)) & x <= (f + d + c + (b/2))){
    mf_c = 1 - ((x - (f + d + c - (b/2))) / (b + delta))
  }
  else if(x > (f + d + c + (b/2)) & x <= 1){
    mf_c = 0
  }
  return (mf_c)
}
else if(b < c & c < d){
  if(x > 0 & x <= (f + d - (c/2))){
    mf_c = 0
  }
  else if (x > (f + d - (c/2)) & x <= (f + d + (c/2))){
    mf_c = 1 + ((x - (f + d + (c/2))) / (c + delta))
  }
  else if(x > (f + d + (c/2)) & x <= (f + d + c - (b/2))){
    mf_c = 1
  }
  else if(x > (f + d + c - (b/2)) & x <= (f + d + c + (b/2))){
    mf_c = 1 - ((x - (f + d + c - (b/2))) / (b + delta))
  }
  else if(x > (f + d + c + (b/2)) & x <= 1){
    mf_c = 0
  }
  return (mf_c)
}
else if(d < c & c < b){
  if(x > 0 & x <= (f + (d/2))){
    mf_c = 0
  }
  else if (x > (f + (d/2)) & x <= (f + (3*d/2))){
    mf_c = 1 + ((x - (f + (3*d/2))) / (d + delta))
  }
  else if(x > (f + (3*d/2)) & x <= (f + d + (c/2))){
    mf_c = 1
  }
  else if(x > (f + d + (c/2)) & x <= (f + d + (3*c/2))){
    mf_c = 1 - ((x - (f + d + (c/2))) / (c + delta))
  }
  else if(x > (f + d + (3*c/2)) & x <= 1){
    mf_c = 0
  }
  return (mf_c)
}
return (mf_c)
}
else{
  if (x>=0 & x<=(f+d-(c/2))){
    mf_c<-0
  } else if (x>({f+d-(c/2)} & x<>({f+d+(c/2)})){
    mf_c<-1+{(x-{f+d+(c/2)})/(c+delta))
  } else if (x>({f+d+(c/2)} & x<i>({f+d+3*(c/2)})){
    mf_c<-1-{(x-{f+d+(c/2)})/(c+delta))
  } else{
    mf_c<-0
  }
  return (mf_c)
}
}

ev_c<-function(iprg){
  a<-iprg[1]
  b<-iprg[2]
  c<-iprg[3]
  d<-iprg[4]
  f<-iprg[5]
  if (c>=max(b,d)){
    ev_c<-((12*c^2+24*c*d+24*c*f+b^2-d^2)/((24*c)+delta)
    return (ev_c)
  } else if (b<c & c<d){
    ev_c<-((11*c^2+24*c*d+24*c*f+b^2)/((24*c)+delta)
    return (ev_c)
  } else if (d<c & c<b){
    ## error in denominator ev_c<-((13*c^2+24*c*d+24*c*f-d^2)/(24*(b+delta))
    ev_c<-((13*c^2+24*c*d+24*c*f-d^2)/((24*c)+delta)
    return (ev_c)
  } else{
    ev_c<-((2*f+2*d+c)/2
    return (ev_c)
  }
}

## for d
mf_d<-function(x,iprg){
  a<-iprg[1]
  b<-iprg[2]
  c<-iprg[3]
  d<-iprg[4]
  f<-iprg[5]
  if (d>=max(c,2*f)){
    if (x>=0 & x<=(2*f)){
      mf_d<-1+(x-2*f)/((2*f)+delta)
    } else if (x>({2*f} & x<i>({f+d-(c/2)})){
      mf_d<-1
    } else if (x>({f+d-(c/2)} & x<i>({f+d+(c/2)})){
      mf_d<-1-(x-{f+d+(c/2)})/(c+delta)
    } else{
      mf_d<-0
    }
  }
return (mf_d)
} else if(c<d & d<(2*f))
if(x>0 & x<=f-(d/2))
    mf_d<0
} else if (x>(f-(d/2)) & x<=(f+(d/2))){
    mf_d<1+
} else if(x>(f+(d/2)) & x<=(f-(c/2))){
    mf_d<1
} else if(x>(f+(d/2)) & x<=(f+(3*d/2))){
    mf_d<1-
} else{
    mf_d<0
}
return (mf_d)
} else if((2*f)<d & d<c){
if(x>0 & x<=2*f){
    mf_d<1+
} else if (x>(2*f) & x<=(f+(d/2))){
    mf_d<1
} else if(x>(f+(d/2)) & x<=(f+(3*d/2))){
    mf_d<1-
} else{
    mf_d<0
}
return (mf_d)
} else{
    if(x>0 & x<=(f-(d/2))){
        mf_d<0
    } else if (x>(f-(d/2)) & x<=(f+(d/2))){
        mf_d<1+
    } else if(x>(f+(d/2)) & x<=(f+(3*d/2))){
        mf_d<1-
    } else{
        mf_d<0
    }
return (mf_d)
}

a<iprg[1]
b<iprg[2]
c<iprg[3]
d<iprg[4]
f<iprg[5]
if(d=max(c,2*f)){
ev_d<-(12*d^2+24*f*d-4*f^2+c^2)/((24*d)+delta)
return (ev_d)
} else if(c<d & d<(2*f)){
ev_d<-(11*d^2+24*f*d+f^2+c^2)/((24*d)+delta)
return (ev_d)
} else if(2*f<d & d<c){
ev_d<-(3*d^2+24*d*f-4*f^2)/((24*d)+delta)
return (ev_d)
else{
    ev_d<={(2*f)+d)/2
    return (ev_d)
}
}

## for f
mf_f<+function(x,iprg){
a<-iprg[1]
b<-iprg[2]
c<-iprg[3]
d<-iprg[4]
f<-iprg[5]
if((2*f)<d){
    if(x>0 & x<=(2*f)){
        mf_f<-1-(x/((2*f)+delta))
    } else{
        mf_f<-0
    }
    return (mf_f)
} else{
    if(x>0 & x<=(f-(d/2))){
        mf_f<-1
    } else if (x>=(f-(d/2)) & x<=(f+(d/2))){
        mf_f<-1-((x-(f-(d/2)))/(d+delta))
    } else{
        mf_f<-0
    }
    return (mf_f)
}
}

## M (Nx5) matrix construction
M<-function(scores,iprg,N){
    # N: the number of quizzes in a course
    M<-matrix(,nrow=N,ncol=5)
    for(i in 1:N){
        M[i,1]<-mf_a(scores[i],iprg)
        M[i,2]<-mf_b(scores[i],iprg)
        M[i,3]<-mf_c(scores[i],iprg)
        M[i,4]<-mf_d(scores[i],iprg)
        M[i,5]<-mf_f(scores[i],iprg)
    }
}

## for f
mf_f<-+function(x,iprg){
a<-iprg[1]
b<-iprg[2]
c<-iprg[3]
d<-iprg[4]
f<-iprg[5]
if((2*f)<d){
    if(x>0 & x<=(2*f)){
        mf_f<-1-(x/((2*f)+delta))
    } else{
        mf_f<-0
    }
    return (mf_f)
} else{
    if(x>0 & x<=(f-(d/2))){
        mf_f<-1
    } else if (x>=(f-(d/2)) & x<=(f+(d/2))){
        mf_f<-1-((x-(f-(d/2)))/(d+delta))
    } else{
        mf_f<-0
    }
    return (mf_f)
}
}

## for ev
ev_d<-+function(x,iprg){
    if((2*f)+d)/2
    return (ev_d)
}

## M (Nx5) matrix construction
M<-function(scores,iprg,N){
    # N: the number of quizzes in a course
    M<-matrix(,nrow=N,ncol=5)
    for(i in 1:N){
        M[i,1]<-mf_a(scores[i],iprg)
        M[i,2]<-mf_b(scores[i],iprg)
        M[i,3]<-mf_c(scores[i],iprg)
        M[i,4]<-mf_d(scores[i],iprg)
        M[i,5]<-mf_f(scores[i],iprg)
    }
}
CHATZISTAVROU KYRIAKI

final_score <- function(w, t, scores, N) {
  sum <- 0
  for (i in 1:N) {
    sum <- sum + (w[i] * t[i])
  }
  return(sum)
}

shinyServer(
  function(input, output, session) {
    observe(
      val <- 100 - (input$perc_a + input$perc_c + input$perc_d + input$perc_f)
      updateSliderInput(session, "perc_b", value = val,
        min = 0, max = 100, step = 1)
    )
    pA <- reactive(input$perc_a)
    pB <- reactive(input$perc_b)
    pC <- reactive(input$perc_c)
    pD <- reactive(input$perc_d)
    pF <- reactive(input$perc_f)
    vectorGrades <- reactive(as.numeric(unlist(as.list(strsplit(input$grades, ",")))))
    vectorWeights <- reactive({
      if (input$eqWeights == TRUE) {
        seq(from = 100, to = 100, length.out = length(vectorGrades()))/length(vectorGrades())
      } else {
        as.numeric(unlist(as.list(strsplit(input$weights, ","))))
      }
    })
    iprg <- reactive((c(pA(), pB(), pC(), pD(), pF()))/100)
    w <- reactive((vectorWeights())/100)
    ## defuzzify
    m <- reactive(M(vectorGrades()/100, iprg(), length(vectorGrades())))
    # vector of expected values
    ev <- reactive(c(ev_a(iprg()), ev_b(iprg()), ev_c(iprg()), ev_d(iprg()), ev_f(iprg())))
    t <- reactive(m() %*% ev())
    final <- reactive(final_score(w(), t(), vectorGrades() / 100, length(vectorGrades())))
    # prints the percentages the user entered
    output$inputPercentages <- renderText({
      if ((input$perc_a + input$perc_b + input$perc_c + input$perc_d + input$perc_f) != 100) {
        paste0("The sum of percentages should equal to 100. Please correct them in order to continue")
      } else {
        paste0("Percentages: ",
          "A: ", pA(), ",%",
          "B: ", pB(), ",%",
          "C: ", pC(), ",%",
          "D: ", pD(), ",%",
          "F: ", pF(), ",%")
      }
    })
    # prints the grades the user entered
output$inputGrades<-renderText({paste0(vectorGrades())})
output$inputWeights<-renderText({
  if(input$eqWeights==FALSE){
    if(length(vectorGrades())!=length(vectorWeights())){
      paste0("You entered ", length(vectorGrades()),
             " grades and ", length(vectorWeights()),
             " weights. The number of grades and weights should be equal.")
    }
    else if(sum(vectorWeights())<100){
      paste0("The sum of weights should be equal to 100.",
             "The sum of the weights that you entered is ",
             sum(vectorWeights()),
             ".You should distribute ", 100-sum(vectorWeights()),
             " more weights.")
    }
    else if(sum(vectorWeights())>100){
      paste0("The sum of weights should be equal to 100.",
             "The sum of the weights that you entered is ",
             sum(vectorWeights()),
             ".You should distribute ", sum(vectorWeights())-100,
             " less weights.")
    }
    else{paste0(vectorWeights())
    }
  }
  else{paste0("All the grades are equal.")
})
output$inputResults<-renderText({paste0(ev())})
output$inputResults1<-renderText({paste0(iprg())})
output$inputResults2<-renderText({
  if(sum(vectorWeights())==100 &
    (input$perc_a + input$perc_b + input$perc_c + input$perc_d + input$perc_f)==100
    & length(vectorGrades())==length(vectorWeights())){}
  else if ((input$perc_a + input$perc_b + input$perc_c + input$perc_d + input$perc_f)!=100){
    paste0("You should first correct the percentages!")
  }
  else if(length(vectorGrades())!=length(vectorWeights())){
    paste0("You should first correct the number of grades and weights in order to be equal!")
  }
  else{ paste0("You should first correct the weights!")
  }
})
output$realGrade<-renderText({
  if(sum(vectorWeights())==100 &
    (input$perc_a + input$perc_b + input$perc_c + input$perc_d + input$perc_f)==100
    & length(vectorGrades())==length(vectorWeights())){
    paste0(sum(vectorGrades())*vectorWeights()/100)/length(vectorGrades))
  }
  else if ((input$perc_a + input$perc_b + input$perc_c + input$perc_d + input$perc_f)!=100){
    paste0("You should first correct the percentages!")
  }
  else if(length(vectorGrades())!=length(vectorWeights())){
    paste0("You should first correct the number of grades and weights in order to be equal!")
  }
  else{
    paste0("You should first correct the weights!")
  }
})
function

```r
# iprg: vector with the ideal percentages of receiving grades A,B,C,D,F
# the instructor sets the values of the vector
data<-as.matrix(read.table("syntheticDataMultipleGrades.txt"))[2:101,1:2]
dataset<-matrix(as.numeric(data), nrow=nrow(data), ncol=ncol(data))

FuzzyMatrix <- FuzzyGradeNew(delta=0,
a=0.2, b=0.2, c=0.20, d=0.20, f=0.2,
  score_table = dataset,
  w=c(1))

#There is only one function the FuzzyGradeNew
#a,b,c,d,f: predetermined percentages of the ideal learners recieving the grades A,B,C,D,F
#score_table: the table with the grades
#w: weights of quizzes
"FuzzyGradeNew" <- function (delta=0,
a=0.15, b=0.35, c=0.20, d=0.20, f=0.10,
  score_table = as.matrix(ExampleDataset2),
  w=c(1))
{

#ideal percentages
iprg <- c(a,b,c,d,f)

### Define the number of learners
n <- nrow(score_table) #number of learners
m <- ncol(score_table) #number of tests

#w<-c(0.1,0.15,0.2,0.25,0.3)
## membership functions and expected values
## xEµ [0,1]
### for a
mf_a<-function(x,iprg){
a<-iprg[1]
b<-iprg[2]
c<-iprg[3]
d<-iprg[4]
f<-iprg[5]
if((2*a)<=b){
  if(x>=0 & x<=(1-(2*a))){
    mf_a<-0
  } else{
    mf_a<-1+((x-(1-(2*a)))/(b+delta))
  }
  return (mf_a)
} else{
  if(x>=0 & x<=a-(b/2)){
    mf_a<-0
  } else if (x>(a-(b/2)) & x<=1-a+(b/2)){
    mf_a<-1+((x-(1-a+(b/2)))/(b+delta))
  } else if (x>1-a+(b/2) & x<=1){
    mf_a<-1+((x-1)/(delta))
  } else if(x>1){
    mf_a<-0
  }
  return (mf_a)
}
```

```
else{
    mf_a<-1
}

return (mf_a)
}

ev_a<-function(iprg){
a<-iprg[1]
b<-iprg[2]
c<-iprg[3]
d<-iprg[4]
f<-iprg[5]
if(2*a<=b){
ev_a<-(-3+2*a)/3
return (ev_a)
}
else{
ev_a<-((24*a-12*a^2-b^2)/((24*a)+delta)
return (ev_a)
}
}

## for b
mf_b<-function(x,iprg){
a<-iprg[1]
b<-iprg[2]
c<-iprg[3]
d<-iprg[4]
f<-iprg[5]
if(b>=max((2*a),c)){
if(x>=0 & x<=(f+d+(c/2))){
    mf_b<-0
}
else if (x>(f+d+(c/2)) & x<=(f+d+(3*c/2))){
    mf_b<-1+((x-(f+d+(3*c/2)))/c+delta)
}
else if (x>(f+d+(3*c/2)) & x<=(1-a)){
    mf_b<-1
}
else{
    mf_b<-1-(x-(1-2*a))/((2*a+delta))
}
return (mf_b)
}
else if((2*a)<b & b<c){
if(x>=0 & x<=(1-a-(3*b/2))){
    mf_b<-0
}
else if (x>(1-a-(3*b/2)) & x<=(1-a-(b/2))){
    mf_b<-1+((x-(1-a-(b/2)))/(b+delta))
}
else if(x>(1-a-(b/2)) & x<=(1-2*a)){
    mf_b<-1
}
else{
    mf_b<-1-(x-(1-2*a))/(2*a+delta))
}
return (mf_b)
}
else if(c<b & b<(2*a)){
if(x>=0 & x<=(f+d+(c/2)){

```plaintext
mf_b<0

else if (x>(f+d+(c/2)) & x<=(f+d+(3*c/2)))
    mf_b<1+(x-(f+d+(3*c/2)))/(c+delta)

else if (x>(f+d+3*(c/2)) & x<=(f+d+c+(b/2)))
    mf_b<1

else if(x>(f+d+c+(b/2)) & x<=(f+d+c+(3*b/2))){
    mf_b<1-((x-(f+d+c+(b/2)))/(b+delta))
}
else {
    mf_b<0
}

return (mf_b)
}

else{
    if(x<0 & x<=(1-a-(3*b/2))){
        mf_b<0
    }
    else if (x>(1-a-(3*b/2)) & x<=(1-a-(b/2))){
        mf_b<1+((x-(1-a-(b/2)))/(b+delta))
    }
    else if(x>(1-a-(b/2)) & x<=(1-a+(b/2))){
        mf_b<1-((x-(1-a-(b/2)))/(b+delta))
    }
    else{
        mf_b<0
    }

    return (mf_b)
}
}
ev_b<function(iprg){
a<-iprg[1]
b<-iprg[2]
c<-iprg[3]
d<-iprg[4]
f<-iprg[5]
if(b>=max((2*a),c)){
ev_b<-(4*a^2-24*a*b+12*b^2+24*b-c^2)/((24*b)+delta)
return (ev_b)
}
else if((2*a)<b & b<c){
ev_b<-(4*a^2-24*a*b+13*b^2+24*b)/(24*b)+delta)
return (ev_b)
}
else if(c<b & b<(2*a)){
ev_b<-(13*b^2+24*b*c+24*b*d+24*b*f-c^2)/((24*b)+delta)
return (ev_b)
}
else{
    ev_b<-(2-2*a-b)/2
    return (ev_b)
}
}
```

## for c
mf_c<function(x,iprg){
a<-iprg[1]
b<-iprg[2]
c<-iprg[3]
d<-iprg[4]
f<-iprg[5]
if(c>=max(b,d)){
    if(x>0 & x<=(f+(d/2))){
        mf_c<0
    }
    else if (x>(f+(d/2)) & x<=(f+(3*d/2))){
```
\[
\text{mf}_c = \begin{cases} 
1 + \frac{x-(f+(3d/2))}{(d+\delta)} & \text{if } x > (f+(3d/2)) \text{ and } x \leq (f+d+c-(b/2)) \\
1 & \text{if } x > (f+d+c-(b/2)) \text{ and } x \leq (f+d+(c/2)) \\
1 - \frac{x-(f+d+c-(b/2))}{(b+\delta)} & \text{if } x > (f+d+(c/2)) \text{ and } x \leq (f+d+c+(b/2)) \\
0 & \text{if } x > (f+d+c+(b/2)) \text{ and } x \leq 1
\end{cases}
\]

return (mf_c)
}
```c
return (mf_c)
}
}
ev_c<-function(iprg) {
a<-iprg[1]
b<-iprg[2]
c<-iprg[3]
d<-iprg[4]
f<-iprg[5]
if(c>=max(b,d)) {
  ev_c<-((12*c^2+24*c*d+24*c*f+b^2-d^2))/((24*c)+delta)
  return (ev_c)
}
else if(b<c & c<d) {
  ev_c<-((11*c^2+24*c*d+24*c*f+b^2))/((24*c)+delta)
  return (ev_c)
}
else if(d<c & c<b) {
  ev_c<-(13*c^2+24*c*d+24*c*f-d^2)/(24*(b+delta))
  ev_c<-(13*c^2+24*c*d+24*c*f-d^2)/((24*c)+delta)
  return (ev_c)
}
else {
  ev_c<-((2*f+2*d+c)/2)
  return (ev_c)
}
}
## for d
mf_d<-function(x,iprg) {
a<-iprg[1]
b<-iprg[2]
c<-iprg[3]
d<-iprg[4]
f<-iprg[5]
if(d>=max(c,2*f)) {
  if(x>=0 & x<=(2*f)) {
    mf_d<1+(x-2*f)/((2*f)+delta)
  }
  else if (x>(2*f) & x<=(f+d-(c/2))) {
    mf_d<0
  }
  else if(x>(f+d-(c/2)) & x<=(f+d+(c/2))) {
    mf_d<1-
    (x-(f+d-(c/2)))/(c+delta)
  }
  else {
    mf_d<0
  }
  return (mf_d)
}
else if(c<d & d<2*f) {
  if(x>=0 & x<=(f-(d/2))) {
    mf_d<1-
    (x-(f+(d/2)))/(d+delta)
  }
  else if (x>(f-(d/2)) & x<=(f+(d/2))) {
    mf_d<0
  }
  else if(x>(f+(d/2)) & x<=(f+d-(c/2))) {
    mf_d<1-
    (x-(f+(d/2)))/((c+delta)
  }
  else {
    mf_d<0
  }
  return (mf_d)
}
else if(2*f<d & d<c) {
}
```

if(x>=0 & x<=2*f){
  mf_d<-1+((x-2*f)/(2*f)+delta))
} else if (x>(2*f) & x<=(f+(d/2))){
  mf_d<-1
} else if (x>(f+(d/2)) & x<=(f+(3*d/2))){
  mf_d<-1-((x-(f+(d/2)))/(d+delta))
} else{
  mf_d<-0
}
return (mf_d)
}

else{
if(x>=0 & x<(f-(d/2))){
  mf_d<-0
} else if (x>(f-(d/2)) & x<=(f+(d/2))){
  mf_d<-1+((x-(f+(d/2)))/(d+delta))
} else if(x>(f+(d/2)) & x<=(f+(3*d/2))){
  mf_d<-1-((x-(f+(d/2)))/(d+delta))
} else{
  mf_d<-0
}
return (mf_d)
}

ev_d<function(iprg){
a<-iprg[1]
b<-iprg[2]
c<-iprg[3]
d<-iprg[4]
f<-iprg[5]
if(d>=max(c,(2*f))){
  ev_d<-(12*d^2+24*f*d-4*f^2+c^2)/((24*d)+delta)
  return (ev_d)
} else if(c<d & d<(2*f)){
  ev_d<-(11*d^2+24*d*f+c^2)/((24*d)+delta)
  return (ev_d)
} else if(2*f<d & d<c){
  ev_d<-(3*d^2+24*d*f-4*f^2)/((24*d)+delta)
  return (ev_d)
} else{
  ev_d<-(2*f)+d/2
  return (ev_d)
}
}

## for f
mf_f<function(x,iprg){
a<-iprg[1]
b<-iprg[2]
c<-iprg[3]
d<-iprg[4]
f<-iprg[5]
if((2*f)<=d){
  if(x>=0 & x<=(2*f)){
    mf_d<-1+((x-2*f)/(2*f)+delta))
  } else if (x>(2*f) & x<=(f+(d/2))){
    mf_d<-1
  } else if (x>(f+(d/2)) & x<=(f+(3*d/2))){
    mf_d<-1-((x-(f+(d/2)))/(d+delta))
  } else{
    mf_d<-0
  }
  return (mf_d)
} else{
  if(x>=0 & x<(f-(d/2))){
    mf_d<-0
  } else if (x>(f-(d/2)) & x<=(f+(d/2))){
    mf_d<-1+((x-(f+(d/2)))/(d+delta))
  } else if(x>(f+(d/2)) & x<=(f+(3*d/2))){
    mf_d<-1-((x-(f+(d/2)))/(d+delta))
  } else{
    mf_d<-0
  }
  return (mf_d)
}
}
mf_f<-1-\left\{ \frac{x}{(2*f)+delta} \right\}

\text{else}\{
    mf_f<-0
\}

\text{return (mf_f)}

\text{else}\{
    \text{if}(x>0 \ \& \ x<(f-(d/2)))\{
        mf_f<-1
    \}
    \text{else if} \ (x>(f-(d/2)) \ \& \ x<(f+(d/2)))\{
        mf_f<-1-\left\{ \frac{x-(f-(d/2))}{d+delta} \right\}
    \}
    \text{else}\{
        mf_f<-0
    \}
\}

\text{return (mf_f)}

ev_f<-\text{function(iprg)}\
    a<-iprg[1]\
    b<-iprg[2]\
    c<-iprg[3]\
    d<-iprg[4]\
    f<-iprg[5]\
    \text{if}((2*f)<=d)\{
        ev_f<-\frac{(2*f)\times 3}{3}
    \}
    \text{return (ev_f)}
    \text{else}\{
        ev_f<-\frac{(d^2+12*f^2)}{(24*f)+delta}
    \}
    \text{return (ev_f)}
}

# scores is a vector containing the row scores of the quizzes in a course
# there are n score vectors one for each learner
#scores<-c(x1,x2,x3,x4,x5)

## M (mx5) matrix construction
M<-\text{function(scores,iprg,m)}\
    # N: the number of quizzes in a course
    M <-\begin{matrix} \text{matrix} & ,nrow=m,ncol=5 \end{matrix}\
    \text{for}(i \ \text{in} \ 1:m)\{
        M[i,1]<-mf_a(scores[i],iprg)\
        M[i,2]<-mf_b(scores[i],iprg)\
        M[i,3]<-mf_c(scores[i],iprg)\
        M[i,4]<-mf_d(scores[i],iprg)\
        M[i,5]<-mf_f(scores[i],iprg)
    }\
    \text{return (M)}

)

t_table <- \text{matrix}(sample(0:100,n*m, replace=TRUE),nrow = n, ncol = m)

# vector of expected values
ev<-c(ev_a(iprg),ev_b(iprg),ev_c(iprg),ev_d(iprg),ev_f(iprg))

## defuzzify
\text{for} \ (i \ \text{in} \ 1:n)\{
    t_table[i,]<-M(score_table[i,],iprg,m)\times ev
\}

final_w_score_vec <- function(w, t, n, m) {
  sum <- rep(0, n)
  for (j in 1:n) {
    sum[j] <- 0
    for (i in 1:m) {
      sum[j] <- sum[j] + (w[i] * t[j, i])
    }
  }
  return(sum)
}

final_score_vec <- function(t, n, m) {
  sum <- rep(0, n)
  for (j in 1:n) {
    sum[j] <- 0
    for (i in 1:m) {
      sum[j] <- sum[j] + t[j, i]
    }
  }
  return(sum/m)
}

fuzzy_w_grade <- final_w_score_vec(w, t_table, n, m)
real_w_grade <- final_w_score_vec(w, score_table, n, m)

fuzzy_grade <- final_score_vec(t_table, n, m)
real_grade <- final_score_vec(score_table, n, m)

fuzzy_w_ranking <- rank(fuzzy_w_grade, ties.method = "average")
real_w_ranking <- rank(real_w_grade, ties.method = "average")
real_ranking <- rank(real_grade, ties.method = "average")

fuzzy_grade_LingGr <- vector("numeric", n)
real_grade_LingGr <- vector("numeric", n)

for (i in 1:n) {
  if (fuzzy_grade[i] < 0.2) {
    fuzzy_grade_LingGr[i] <- c("F")
  } else if ((fuzzy_grade[i] < 0.4) & (fuzzy_grade[i] >= 0.2)) {
    fuzzy_grade_LingGr[i] <- c("D")
  } else if ((fuzzy_grade[i] < 0.6) & (fuzzy_grade[i] >= 0.4)) {
    fuzzy_grade_LingGr[i] <- c("C")
  } else if ((fuzzy_grade[i] < 0.8) & (fuzzy_grade[i] >= 0.6)) {
    fuzzy_grade_LingGr[i] <- c("B")
  } else {
    fuzzy_grade_LingGr[i] <- c("A")
  }
}

for (i in 1:n) {
  if (real_grade[i] < 0.2) {
    real_grade_LingGr[i] <- c("F")
  } else if ((real_grade[i] < 0.4) & (real_grade[i] >= 0.2)) {
    real_grade_LingGr[i] <- c("D")
  } else if ((real_grade[i] < 0.6) & (real_grade[i] >= 0.4)) {
    real_grade_LingGr[i] <- c("C")
  } else if ((real_grade[i] < 0.8) & (real_grade[i] >= 0.6)) {
    real_grade_LingGr[i] <- c("B")
  } else {
    real_grade_LingGr[i] <- c("A")
  }
}
real_grade_LingGr[i] <- c("C")
} else if ((real_grade[i]<0.80) & (real_grade[i]>=0.60)){
  real_grade_LingGr[i] <- c("B")
} else
  real_grade_LingGr[i] <- c("A")
}

resultLing <- data.frame(fuzzy_grade_LingGr,real_grade_LingGr)
resultMatrix <-cbind(
  fuzzy_w_grade, real_w_grade,
  fuzzy_grade, real_grade,
  fuzzy_w_ranking,real_w_ranking,
  fuzzy_ranking,real_ranking
)
result <- data.frame(resultMatrix,resultLing)
colnames(result) <- c(
  #fuzzy_w_grade, real_w_grade,
  "f_w_g","r_w_g",
  "f_g","r_g",
  #fuzzy_w_rank,real_w_rank,
  "f_w_rank","r_w_rank",
  "f_rank","r_rank",
  "f_g_Ling","r_g_Ling")

membership_fun<-function(score,iprg){
  vec_a<-rep(NA,length(score))
  for (i in 1:length(score)){
    vec_a[i]<-mf_a(score[i],iprg)
  }

  vec_b<-rep(NA,length(score))
  for (i in 1:length(score)){
    vec_b[i]<-mf_b(score[i],iprg)
  }

  vec_c<-rep(NA,length(score))
  for (i in 1:length(score)){
    vec_c[i]<-mf_c(score[i],iprg)
  }

  vec_d<-rep(NA,length(score))
  for (i in 1:length(score)){
    vec_d[i]<-mf_d(score[i],iprg)
  }

  vec_f<-rep(NA,length(score))
  for (i in 1:length(score)){
    vec_f[i]<-mf_f(score[i],iprg)
  }

  mf_matrix<-cbind(vec_a,vec_b,vec_c, vec_d,vec_f)
  return(mf_matrix)
}

mf_mat<-matrix(NA,nrow=n,ncol=5*m+m)
for (i in 1:m){
  if (i==1){
    temp_mf_mat<-round(membership_fun(score_table[,i],iprg),3)
    mf_mat<-cbind(temp_mf_mat, rowSums(temp_mf_mat))
  }
  else{
    temp_mf_mat<-round(membership_fun(score_table[,i],iprg),3)
    mf_mat<-cbind(mf_mat,temp_mf_mat, rowSums(temp_mf_mat))
  }
}
list(result=result,ev=ev,mf_mat=mf_mat)
Appendix B – Results of Testing

Case Study 1
Case Study 2

Distribution of Real Grades

Distribution of Fuzzy Grades

Density of Real Grades

Density of Fuzzy Grades

Real Grade

Fuzzy Grade
Case Study 3
Case Study 5
Case Study 6
Case Study 7
Case Study 8

[Graphs showing distribution and density of real and fuzzy grades]
Case Study 9
Case Study 10
Case Study 11

- Distribution of Real Grades
  - Count
  - Grade: 0.00, 0.25, 0.50, 0.75, 1.00
  - Grades: A, B, C, D, F

- Distribution of Fuzzy Grades
  - Count
  - Grade: 0.25, 0.50, 0.75, 1.00
  - Grades: A, B, C, D

- Density of Real Grades
  - Density
  - Grade: 0.00, 0.25, 0.50, 0.75, 1.00

- Density of Fuzzy Grades
  - Density
  - Grade: 0.2, 0.4, 0.6, 0.8, 1.0

- Real Grade vs. Fuzzy Grade
  - Real Grade: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
  - Fuzzy Grade: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
Case Study 12
Case Study 13

[Graphs showing distributions of real and fuzzy grades, along with their densities and scatter plot of real vs. fuzzy grades.]
Case Study 14
Case Study 15
Case Study 16
library(shiny)
shinyUI(pageWithSidebar(
    headerPanel("Ranking"),
    sidebarPanel(
        textInput(inputId="cde", "Insert your code to see your results", "43"),
        helpText("Your code should be between 1-1044"),
        fluidPage(
            fluidRow(
                column(12,
                    h4("Your Grades are: "),
                    h5(textOutput("grade1")),
                    h5(textOutput("grade2")),
                    h5(textOutput("grade3"))
                )
            ),
            fluidRow(
                column(12,
                    h4("Your GPA is: "),
                    h5(textOutput("gpa"))
                )
            ),
            fluidRow(
                column(12,
                    h4("Your Rank is: "),
                    h5(textOutput("rank1")),
                    h5(textOutput("rank2")),
                    h5(textOutput("rank3")),
                    h5(textOutput("rankgpa"))
                )
            ),
            width=3
        ),
        mainPanel{
            fluidPage(
                fluidRow(
                    column(6,
                        plotOutput("G1plot",width="100%",height="300px")
                    ),
                    column(6,
                        plotOutput("G2plot",width="100%",height="300px")
                    )
                ),
                fluidRow(
                    column(6,
                        plotOutput("G3plot",width="100%",height="300px")
                    ),
                    column(6,
                        plotOutput("GPAplot",width="100%",height="300px")
                    )
                ),
                fluidRow(
                    column(12,
                        plotOutput("Gplot",width="100%",height="400px")
                    )
                )
            )
        )
    )
))
library(shiny)
interactive() == TRUE

library(ggplot2)
#setwd("../Desktop")

#read data
data <- read.csv("students.csv", sep=";")

#create the vector G1, G2, G3 and GPA
G1 <- 5 * data$G1
G2 <- 5 * data$G2
G3 <- 5 * data$G3
GPA <- format(((G1 + G2 + G3) / 3), digits = 3)

#mine the grades
grades <- function(cd){
  cd <- as.integer(unlist(cd))
  myG1 <- G1[cd]
  myG2 <- G2[cd]
  myG3 <- G3[cd]
  return(c(myG1, myG2, myG3))
}

#calculate the GPA
myGPA <- function (cd){
  cd <- as.integer(unlist(cd))
  return(GPA[cd])
}

#find the ranking
numG1 <- function (cd){return(which(unique(sort(G1, decreasing = T)) == grades(cd)[1]))}
numG2 <- function (cd){return(which(unique(sort(G2, decreasing = T)) == grades(cd)[2]))}
numG3 <- function (cd){return(which(unique(sort(G3, decreasing = T)) == grades(cd)[3]))}
numGPA <- function (cd){return(which(unique(sort(GPA, decreasing = T)) == myGPA(cd)))}

shinyServer(
  function(input, output, session){
    cd <- reactive(as.integer(unlist(input$cde)))
    # display the learner's grades
    output$grade1 <- renderText(paste0("Test 1: ", grades(input$cde)[1], "\%"))
    output$grade2 <- renderText(paste0("Test 2: ", grades(input$cde)[2], "\%"))
    output$grade3 <- renderText(paste0("Test 3: ", grades(input$cde)[3], "\%"))
    
    # display the learner's GPA
    output$gpa <- renderText(paste0(myGPA(input$cde), "\%"))
    
    # display the learner's Rank
    output$rank1 <- renderText(paste0("Test 1: ", numG1(input$cde)))
    output$rank2 <- renderText(paste0("Test 2: ", numG2(input$cde)))
    output$rank3 <- renderText(paste0("Test 3: ", numG3(input$cde)))
`output$rankgpa <- renderText({paste0("GPA: ", numGPA(input$cde))})`

`output$G1plot <- renderPlot({
df <- data.frame(
  name = c("Your Grade", "Mean"),
  val = c(grades(input$cde)[1], mean(G1)))
ggplot(data, aes(5*G1)) +
  geom_histogram(binwidth=1.5, color="gray", fill="gray") +
  labs(title = "Test 1 Scores", x = "Grades") +
  geom_vline(data = df,
    aes(xintercept = val,
      color = name),
    show_guide = TRUE)
})`

`# plot the histogram of G2`
`output$G2plot <- renderPlot({
  df <- data.frame(
    name = c("Your Grade", "Mean"),
    val = c(grades(input$cde)[2], mean(G2)))
ggplot(data, aes(5*G2)) +
  geom_histogram(binwidth=1.5, color="gray", fill="gray") +
  labs(title = "Test 2 Scores", x = "Grades") +
  geom_vline(data = df,
    aes(xintercept = val,
      color = name),
    show_guide = TRUE)
})`

`# plot the histogram of G3`
`output$G3plot <- renderPlot({
  df <- data.frame(
    name = c("Your Grade", "Mean"),
    val = c(grades(input$cde)[3], mean(G3)))
ggplot(data, aes(5*G3)) +
  geom_histogram(binwidth=1.5, color="gray", fill="gray") +
  labs(title = "Test 3 Scores", x = "Grades") +
  geom_vline(data = df,
    aes(xintercept = val,
      color = name),
    show_guide = TRUE)
})`

`# plot the histogram of GPA`
`output$GPAplot <- renderPlot({
  df <- data.frame(
    name = c("Your GPA", "Mean GPA"),
    val = c(mean(grades(input$cde)), mean(as.integer(GPA))))
ggplot(data, aes(5*((G1+G2+G3)/3))) +
  geom_histogram(binwidth=1.5, color="gray", fill="gray") +
  labs(title = "Test 1, 2 & 3 Scores", x = "Grades") +
  geom_vline(data = df,
    aes(xintercept = val,
      color = name),
    show_guide = TRUE)
})`

`output$rangePlot <- renderPlot({
  # plot the point range
  d = data.frame(
    test = c("Test 1", "Test 2", "Test 3", "GPA"),
    yourGrade = c(grades(input$cde)[1],
                  grades(input$cde)[2],
                  grades(input$cde)[3],
                  as.integer(mean(grades(input$cde))))),
    lower = c(min(G1), min(G2), min(G3), min(as.integer(GPA))),
    upper = c(max(G1), max(G2), max(G3), max(as.integer(GPA))))
cuts = data.frame(
    means = c("Mean Test 1", "Mean Test 2", "Mean Test 3", "Mean GPA"),
    val = c(mean(G1), mean(G2), mean(G3), mean(as.integer(GPA))))
ggplot() +
  geom_pointrange(data = d,
                  mapping = aes(x = test, y = yourGrade,
```r
# radar plot
output$Gplot<-
renderPlot({
  var.names <- c("Test 1", "Test 2", "Test 3", "GPA")
  var.order <- seq(1:4)
  # your Grades
  values.a <- c(grades(input$cde)[1],
                grades(input$cde)[2],
                grades(input$cde)[3],
                as.integer(mean(grades(input$cde))))
  # max grades
  values.b <- c(max(G1), max(G2), max(G3), max(as.integer(GPA)))
  # avg grades
  values.c <- c(mean(G1), mean(G2), mean(G3), mean(as.integer(GPA)))
  group.names <- c("your Grades", "max Grades", "AVG Grades")
  df1.a <- data.frame(matrix(c(rep(group.names[1], 4), var.names),
                            nrow = 4, ncol = 2), var.order = var.order, value = values.a)
  df1.b <- data.frame(matrix(c(rep(group.names[2], 4), var.names),
                            nrow = 4, ncol = 2), var.order = var.order, value = values.b)
  df1.c <- data.frame(matrix(c(rep(group.names[3], 4), var.names),
                            nrow = 4, ncol = 2), var.order = var.order, value = values.c)
  df1 <- rbind(df1.a, df1.b, df1.c)
  colnames(df1) <- c("group", "variable.name", "variable.order", "variable.value")
  ggplot(df1, aes(y = variable.value, x = reorder(variable.name, variable.order),
                group = group, colour = group)) +
  coord_polar() + geom_point() + geom_path() +
  labs(title="Your grades in comparison with the maximum & the AVG grades",
       x = "", y="")
})

# 2 sided barplot
output$barPlot<-
renderPlot({
  yourGrade <- c(grades(input$cde)[1],
                grades(input$cde)[2],
                grades(input$cde)[3],
                as.integer(mean(grades(input$cde))))
  meanGrade <- c(mean(G1), mean(G2), mean(G3), mean(as.integer(GPA)))
  d <- data.frame(test = rep(c("Test 1", "Test 2", "Test 3", "GPA"), times=2),
                 grade = rep(x=c("Your Grades", "AVG Grades"),each=4),
                 grade = rep(x=c("Your Grades", "AVG Grades"),each=4),
                 val = c(yourGrade, meanGrade))
  ggplot(d, aes(x = test, y = val, fill = grade)) +
  geom_bar(data = subset(d, grade="Your Grades"),
            stat = "identity", width=0.5) +
  geom_bar(data = subset(d, grade="AVG Grades"),
            stat = "identity",
            position = "identity",
            mapping = aes(y = -val), width=0.5) +
  scale_y_continuous(labels = abs) +
  coord_flip() +
  labs(title="Your grades in comparison with the AVG grades",
       x="", y="")
})
```
x = " ", y="Grade (0-100)"

})

}"}
# Appendix D – Association Rules Mining

## Input Matrix

<table>
<thead>
<tr>
<th>Language</th>
<th>JSP</th>
<th>LVL</th>
<th>grade</th>
<th>grade_g</th>
<th>grade_v</th>
<th>grade_rw</th>
<th>grade_sl</th>
<th>grade_lsp</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>H</td>
<td>1</td>
<td>0.967</td>
<td>1</td>
<td>1</td>
<td>0.93</td>
<td>0.94</td>
<td>NA</td>
</tr>
<tr>
<td>S</td>
<td>T</td>
<td>1</td>
<td>0.9188</td>
<td>0.93</td>
<td>0.94</td>
<td>0.87</td>
<td>0.94</td>
<td>NA</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>1</td>
<td>0.9188</td>
<td>0.87</td>
<td>0.94</td>
<td>0.93</td>
<td>0.94</td>
<td>NA</td>
</tr>
<tr>
<td>S</td>
<td>H</td>
<td>1</td>
<td>0.898</td>
<td>0.86</td>
<td>0.93</td>
<td>0.87</td>
<td>0.94</td>
<td>NA</td>
</tr>
<tr>
<td>G</td>
<td>B</td>
<td>1</td>
<td>0.8818</td>
<td>0.82</td>
<td>0.92</td>
<td>0.86</td>
<td>0.93</td>
<td>NA</td>
</tr>
<tr>
<td>E</td>
<td>T</td>
<td>1</td>
<td>0.8507</td>
<td>0.74</td>
<td>0.92</td>
<td>0.82</td>
<td>0.92</td>
<td>NA</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>1</td>
<td>0.8287</td>
<td>0.74</td>
<td>0.92</td>
<td>0.74</td>
<td>0.92</td>
<td>NA</td>
</tr>
<tr>
<td>E</td>
<td>H</td>
<td>1</td>
<td>0.8201</td>
<td>0.72</td>
<td>0.91</td>
<td>0.74</td>
<td>0.92</td>
<td>NA</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>2</td>
<td>0.85</td>
<td>0.72</td>
<td>0.9</td>
<td>0.72</td>
<td>0.91</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>T</td>
<td>2</td>
<td>0.83</td>
<td>0.72</td>
<td>0.9</td>
<td>0.72</td>
<td>0.9</td>
<td>0.94</td>
</tr>
<tr>
<td>E</td>
<td>H</td>
<td>2</td>
<td>0.83</td>
<td>0.71</td>
<td>0.89</td>
<td>0.72</td>
<td>0.9</td>
<td>0.94</td>
</tr>
<tr>
<td>S</td>
<td>B</td>
<td>2</td>
<td>0.82</td>
<td>0.7</td>
<td>0.87</td>
<td>0.71</td>
<td>0.89</td>
<td>0.93</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>2</td>
<td>0.81</td>
<td>0.7</td>
<td>0.86</td>
<td>0.7</td>
<td>0.87</td>
<td>0.92</td>
</tr>
<tr>
<td>G</td>
<td>H</td>
<td>2</td>
<td>0.81</td>
<td>0.69</td>
<td>0.86</td>
<td>0.7</td>
<td>0.86</td>
<td>0.92</td>
</tr>
<tr>
<td>S</td>
<td>B</td>
<td>2</td>
<td>0.8</td>
<td>0.66</td>
<td>0.86</td>
<td>0.69</td>
<td>0.86</td>
<td>0.92</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
<td>2</td>
<td>0.79</td>
<td>0.66</td>
<td>0.85</td>
<td>0.66</td>
<td>0.86</td>
<td>0.91</td>
</tr>
<tr>
<td>G</td>
<td>T</td>
<td>2</td>
<td>0.8</td>
<td>0.65</td>
<td>0.84</td>
<td>0.66</td>
<td>0.85</td>
<td>1</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>2</td>
<td>0.78</td>
<td>0.64</td>
<td>0.83</td>
<td>0.65</td>
<td>0.84</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>E</td>
<td>T</td>
<td>2</td>
<td>0.76</td>
<td>0.63</td>
<td>0.81</td>
<td>0.64</td>
<td>0.83</td>
<td>0.87</td>
</tr>
<tr>
<td>G</td>
<td>B</td>
<td>2</td>
<td>0.74</td>
<td>0.61</td>
<td>0.8</td>
<td>0.63</td>
<td>0.81</td>
<td>0.86</td>
</tr>
<tr>
<td>G</td>
<td>T</td>
<td>1</td>
<td>0.81</td>
<td>0.72</td>
<td>0.9</td>
<td>0.72</td>
<td>0.91</td>
<td>0.82</td>
</tr>
<tr>
<td>G</td>
<td>H</td>
<td>1</td>
<td>0.79</td>
<td>0.72</td>
<td>0.9</td>
<td>0.72</td>
<td>0.9</td>
<td>0.74</td>
</tr>
<tr>
<td>G</td>
<td>T</td>
<td>2</td>
<td>0.79</td>
<td>0.71</td>
<td>0.89</td>
<td>0.72</td>
<td>0.9</td>
<td>0.74</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>1</td>
<td>0.78</td>
<td>0.7</td>
<td>0.87</td>
<td>0.71</td>
<td>0.89</td>
<td>0.72</td>
</tr>
<tr>
<td>E</td>
<td>T</td>
<td>1</td>
<td>0.77</td>
<td>0.7</td>
<td>0.86</td>
<td>0.7</td>
<td>0.87</td>
<td>0.72</td>
</tr>
<tr>
<td>E</td>
<td>H</td>
<td>1</td>
<td>0.8248</td>
<td>0.81</td>
<td>0.85</td>
<td>0.8</td>
<td>0.84</td>
<td>NA</td>
</tr>
<tr>
<td>G</td>
<td>T</td>
<td>1</td>
<td>0.8162</td>
<td>0.8</td>
<td>0.84</td>
<td>0.79</td>
<td>0.83</td>
<td>NA</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>1</td>
<td>0.8344</td>
<td>0.9</td>
<td>0.83</td>
<td>0.79</td>
<td>0.81</td>
<td>NA</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
<td>1</td>
<td>0.8501</td>
<td>0.9</td>
<td>0.81</td>
<td>0.78</td>
<td>0.91</td>
<td>NA</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>1</td>
<td>0.8429</td>
<td>0.89</td>
<td>0.8</td>
<td>0.78</td>
<td>0.9</td>
<td>NA</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>1</td>
<td>0.8353</td>
<td>0.87</td>
<td>0.79</td>
<td>0.78</td>
<td>0.9</td>
<td>NA</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>1</td>
<td>0.8273</td>
<td>0.86</td>
<td>0.79</td>
<td>0.77</td>
<td>0.89</td>
<td>NA</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>1</td>
<td>0.782</td>
<td>0.72</td>
<td>0.78</td>
<td>0.72</td>
<td>0.91</td>
<td>NA</td>
</tr>
<tr>
<td>S</td>
<td>H</td>
<td>2</td>
<td>0.75</td>
<td>0.72</td>
<td>0.78</td>
<td>0.72</td>
<td>0.9</td>
<td>0.66</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>2</td>
<td>0.75</td>
<td>0.71</td>
<td>0.78</td>
<td>0.72</td>
<td>0.9</td>
<td>0.65</td>
</tr>
<tr>
<td>S</td>
<td>T</td>
<td>2</td>
<td>0.79</td>
<td>0.7</td>
<td>0.77</td>
<td>0.71</td>
<td>0.89</td>
<td>0.9</td>
</tr>
<tr>
<td>S</td>
<td>H</td>
<td>1</td>
<td>0.78</td>
<td>0.7</td>
<td>0.77</td>
<td>0.7</td>
<td>0.87</td>
<td>0.89</td>
</tr>
<tr>
<td>G</td>
<td>T</td>
<td>2</td>
<td>0.74</td>
<td>0.65</td>
<td>0.74</td>
<td>0.68</td>
<td>0.77</td>
<td>0.87</td>
</tr>
<tr>
<td>S</td>
<td>T</td>
<td>2</td>
<td>0.73</td>
<td>0.65</td>
<td>0.74</td>
<td>0.68</td>
<td>0.74</td>
<td>0.86</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
<td>2</td>
<td>0.73</td>
<td>0.64</td>
<td>0.74</td>
<td>0.67</td>
<td>0.74</td>
<td>0.86</td>
</tr>
<tr>
<td>G</td>
<td>S</td>
<td>2</td>
<td>0.72</td>
<td>0.64</td>
<td>0.73</td>
<td>0.73</td>
<td>0.63</td>
<td>0.86</td>
</tr>
<tr>
<td>S</td>
<td>B</td>
<td>2</td>
<td>0.71</td>
<td>0.63</td>
<td>0.73</td>
<td>0.72</td>
<td>0.63</td>
<td>0.85</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
<td>2</td>
<td>0.73</td>
<td>0.63</td>
<td>0.73</td>
<td>0.85</td>
<td>0.62</td>
<td>0.84</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>1</td>
<td>0.73</td>
<td>0.63</td>
<td>0.72</td>
<td>0.84</td>
<td>0.62</td>
<td>0.83</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>H</td>
<td>1</td>
<td>0.78</td>
<td>0.62</td>
<td>0.71</td>
<td>0.83</td>
<td>0.91</td>
<td>0.81</td>
</tr>
<tr>
<td>S</td>
<td>H</td>
<td>2</td>
<td>0.75</td>
<td>0.62</td>
<td>0.7</td>
<td>0.81</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>S</td>
<td>T</td>
<td>2</td>
<td>0.76</td>
<td>0.6</td>
<td>0.7</td>
<td>0.8</td>
<td>0.89</td>
<td>0.79</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>1</td>
<td>0.75</td>
<td>0.6</td>
<td>0.7</td>
<td>0.79</td>
<td>0.87</td>
<td>0.79</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>2</td>
<td>0.75</td>
<td>0.71</td>
<td>0.7</td>
<td>0.79</td>
<td>0.77</td>
<td>0.78</td>
</tr>
<tr>
<td>E</td>
<td>H</td>
<td>2</td>
<td>0.74</td>
<td>0.7</td>
<td>0.69</td>
<td>0.78</td>
<td>0.74</td>
<td>0.78</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>1</td>
<td>0.74</td>
<td>0.7</td>
<td>0.69</td>
<td>0.78</td>
<td>0.74</td>
<td>0.78</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
<td>2</td>
<td>0.71</td>
<td>0.7</td>
<td>0.69</td>
<td>0.78</td>
<td>0.63</td>
<td>0.77</td>
</tr>
<tr>
<td>G</td>
<td>S</td>
<td>2</td>
<td>0.71</td>
<td>0.7</td>
<td>0.68</td>
<td>0.77</td>
<td>0.63</td>
<td>0.77</td>
</tr>
<tr>
<td>G</td>
<td>S</td>
<td>2</td>
<td>0.7</td>
<td>0.69</td>
<td>0.68</td>
<td>0.77</td>
<td>0.62</td>
<td>0.74</td>
</tr>
<tr>
<td>G</td>
<td>S</td>
<td>2</td>
<td>0.7</td>
<td>0.69</td>
<td>0.68</td>
<td>0.74</td>
<td>0.63</td>
<td>0.74</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>2</td>
<td>0.69</td>
<td>0.69</td>
<td>0.67</td>
<td>0.74</td>
<td>0.62</td>
<td>0.74</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>1</td>
<td>0.69</td>
<td>0.68</td>
<td>0.67</td>
<td>0.74</td>
<td>0.62</td>
<td>0.73</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>2</td>
<td>0.71</td>
<td>0.68</td>
<td>0.67</td>
<td>0.73</td>
<td>0.74</td>
<td>0.73</td>
</tr>
<tr>
<td>E</td>
<td>H</td>
<td>1</td>
<td>0.68</td>
<td>0.65</td>
<td>0.67</td>
<td>0.73</td>
<td>0.63</td>
<td>0.73</td>
</tr>
<tr>
<td>G</td>
<td>B</td>
<td>1</td>
<td>0.68</td>
<td>0.65</td>
<td>0.67</td>
<td>0.73</td>
<td>0.63</td>
<td>0.72</td>
</tr>
<tr>
<td>G</td>
<td>B</td>
<td>1</td>
<td>0.68</td>
<td>0.64</td>
<td>0.66</td>
<td>0.72</td>
<td>0.68</td>
<td>0.71</td>
</tr>
<tr>
<td>S</td>
<td>B</td>
<td>1</td>
<td>0.68</td>
<td>0.64</td>
<td>0.66</td>
<td>0.71</td>
<td>0.67</td>
<td>0.7</td>
</tr>
<tr>
<td>S</td>
<td>T</td>
<td>2</td>
<td>0.67</td>
<td>0.63</td>
<td>0.66</td>
<td>0.7</td>
<td>0.67</td>
<td>0.7</td>
</tr>
<tr>
<td>E</td>
<td>T</td>
<td>1</td>
<td>0.67</td>
<td>0.63</td>
<td>0.65</td>
<td>0.7</td>
<td>0.67</td>
<td>0.7</td>
</tr>
<tr>
<td>S</td>
<td>B</td>
<td>1</td>
<td>0.67</td>
<td>0.63</td>
<td>0.65</td>
<td>0.7</td>
<td>0.67</td>
<td>0.7</td>
</tr>
<tr>
<td>S</td>
<td>T</td>
<td>2</td>
<td>0.66</td>
<td>0.62</td>
<td>0.65</td>
<td>0.7</td>
<td>0.67</td>
<td>0.69</td>
</tr>
<tr>
<td>S</td>
<td>T</td>
<td>1</td>
<td>0.66</td>
<td>0.62</td>
<td>0.64</td>
<td>0.69</td>
<td>0.66</td>
<td>0.69</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>1</td>
<td>0.66</td>
<td>0.6</td>
<td>0.64</td>
<td>0.69</td>
<td>0.68</td>
<td>0.69</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.66</td>
<td>0.6</td>
<td>0.63</td>
<td>0.69</td>
<td>0.67</td>
<td>0.68</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>------</td>
<td>-----</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>E</td>
<td>H</td>
<td>1</td>
<td>0.68</td>
<td>0.71</td>
<td>0.63</td>
<td>0.68</td>
<td>0.67</td>
<td>0.68</td>
</tr>
<tr>
<td>E</td>
<td>T</td>
<td>2</td>
<td>0.67</td>
<td>0.7</td>
<td>0.62</td>
<td>0.68</td>
<td>0.67</td>
<td>0.67</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
<td>1</td>
<td>0.67</td>
<td>0.7</td>
<td>0.62</td>
<td>0.67</td>
<td>0.67</td>
<td>0.67</td>
</tr>
<tr>
<td>G</td>
<td>B</td>
<td>2</td>
<td>0.66</td>
<td>0.7</td>
<td>0.6</td>
<td>0.67</td>
<td>0.66</td>
<td>0.67</td>
</tr>
<tr>
<td>G</td>
<td>S</td>
<td>2</td>
<td>0.65</td>
<td>0.69</td>
<td>0.6</td>
<td>0.67</td>
<td>0.6</td>
<td>0.67</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>2</td>
<td>0.67</td>
<td>0.69</td>
<td>0.71</td>
<td>0.67</td>
<td>0.6</td>
<td>0.67</td>
</tr>
<tr>
<td>G</td>
<td>S</td>
<td>1</td>
<td>0.69</td>
<td>0.69</td>
<td>0.7</td>
<td>0.67</td>
<td>0.71</td>
<td>0.66</td>
</tr>
<tr>
<td>E</td>
<td>H</td>
<td>1</td>
<td>0.68</td>
<td>0.69</td>
<td>0.7</td>
<td>0.66</td>
<td>0.7</td>
<td>0.66</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
<td>2</td>
<td>0.66</td>
<td>0.68</td>
<td>0.7</td>
<td>0.66</td>
<td>0.6</td>
<td>0.66</td>
</tr>
<tr>
<td>E</td>
<td>H</td>
<td>2</td>
<td>0.64</td>
<td>0.63</td>
<td>0.7</td>
<td>0.66</td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>E</td>
<td>H</td>
<td>1</td>
<td>0.68</td>
<td>0.63</td>
<td>0.69</td>
<td>0.65</td>
<td>0.71</td>
<td>0.71</td>
</tr>
<tr>
<td>E</td>
<td>T</td>
<td>1</td>
<td>0.67</td>
<td>0.62</td>
<td>0.69</td>
<td>0.65</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>2</td>
<td>0.67</td>
<td>0.62</td>
<td>0.69</td>
<td>0.65</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>E</td>
<td>T</td>
<td>1</td>
<td>0.71</td>
<td>0.83</td>
<td>0.68</td>
<td>0.64</td>
<td>0.7</td>
<td>0.69</td>
</tr>
<tr>
<td>E</td>
<td>H</td>
<td>1</td>
<td>0.7</td>
<td>0.81</td>
<td>0.68</td>
<td>0.64</td>
<td>0.7</td>
<td>0.67</td>
</tr>
<tr>
<td>G</td>
<td>T</td>
<td>1</td>
<td>0.69</td>
<td>0.8</td>
<td>0.68</td>
<td>0.63</td>
<td>0.69</td>
<td>0.66</td>
</tr>
<tr>
<td>G</td>
<td>H</td>
<td>1</td>
<td>0.69</td>
<td>0.79</td>
<td>0.67</td>
<td>0.63</td>
<td>0.69</td>
<td>0.66</td>
</tr>
<tr>
<td>S</td>
<td>H</td>
<td>1</td>
<td>0.69</td>
<td>0.79</td>
<td>0.67</td>
<td>0.63</td>
<td>0.68</td>
<td>0.65</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
<td>2</td>
<td>0.68</td>
<td>0.78</td>
<td>0.67</td>
<td>0.62</td>
<td>0.68</td>
<td>0.62</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
<td>1</td>
<td>0.67</td>
<td>0.78</td>
<td>0.67</td>
<td>0.62</td>
<td>0.68</td>
<td>0.6</td>
</tr>
<tr>
<td>E</td>
<td>S</td>
<td>1</td>
<td>0.67</td>
<td>0.78</td>
<td>0.67</td>
<td>0.6</td>
<td>0.67</td>
<td>0.6</td>
</tr>
<tr>
<td>G</td>
<td>B</td>
<td>1</td>
<td>0.67</td>
<td>0.77</td>
<td>0.66</td>
<td>0.6</td>
<td>0.67</td>
<td>NA</td>
</tr>
<tr>
<td>G</td>
<td>B</td>
<td>2</td>
<td>0.67</td>
<td>0.77</td>
<td>0.66</td>
<td>0.63</td>
<td>0.6</td>
<td>0.67</td>
</tr>
<tr>
<td>S</td>
<td>B</td>
<td>2</td>
<td>0.65</td>
<td>0.74</td>
<td>0.66</td>
<td>0.63</td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>S</td>
<td>H</td>
<td>2</td>
<td>0.69</td>
<td>0.74</td>
<td>0.65</td>
<td>0.62</td>
<td>0.71</td>
<td>0.71</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>E</td>
<td>T</td>
<td>2</td>
<td>0.68</td>
<td>0.74</td>
<td>0.65</td>
<td>0.62</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>2</td>
<td>0.68</td>
<td>0.73</td>
<td>0.65</td>
<td>0.6</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>S</td>
<td>H</td>
<td>1</td>
<td>0.67</td>
<td>0.73</td>
<td>0.64</td>
<td>0.6</td>
<td>0.7</td>
<td>0.69</td>
</tr>
<tr>
<td>G</td>
<td>B</td>
<td>2</td>
<td>0.69</td>
<td>0.73</td>
<td>0.64</td>
<td>0.73</td>
<td>0.7</td>
<td>0.67</td>
</tr>
<tr>
<td>E</td>
<td>H</td>
<td>2</td>
<td>0.69</td>
<td>0.72</td>
<td>0.63</td>
<td>0.73</td>
<td>0.69</td>
<td>0.66</td>
</tr>
</tbody>
</table>

**Association Rules Mining Code**

```r
# load data
setwd("C:/Users/Kiki/Desktop")
data<-read.csv(file = "AR_TABLE.csv", header = TRUE, sep = ";")

ar<-AR_fu(data)
"AR_fu" <- function (data = data) {
  # initialize the sparse matrix
  mat<-c("E","S","G", # language (English, Spanish, German)
         "H","TH","ST","BPL", # job of specific purpose
         "L1","L2", # level (Beginner, Intermediate)
         "G_LP","G_P","G_HP", # overall grade so far (Low Pass, Pass, High Pass)
         "G_V_LP","G_V_P","G_V_HP", # overall grade so far in grammar (Low Pass, Pass, High Pass)
         "G_RW_LP","G_RW_P","G_RW_HP", # overall grade so far in reading/writing (Low Pass, Pass, High Pass)
         "G_SL_LP","G_SL_P","G_SL_HP", # overall grade so far in speaking/listening (Low Pass, Pass, High Pass)
         "G_LSP_A","G_LSP_LP","G_LSP_P","G_LSP_HP") # overall grade so far in language for specific purposes (Absent,Low Pass, Pass, High Pass)

  # initialize the helpers for the sparse matrix construction
  vecCOURSE<-c(NA,NA,NA)
  vecJSP<-c(NA,NA,NA,NA)
  vecLVL<-c(NA,NA)
  vecG<-c(NA,NA,NA)
  vecGG<-c(NA,NA,NA)
  vecGV<-c(NA,NA,NA)
  vecGRW<-c(NA,NA,NA,NA)
  vecGSL<-c(NA,NA,NA)
  vecGLSP<-c(NA,NA,NA,NA)
  for (i in 1:nrow(data)){
    for (j in 1:9){
      # Course
      if(j==1){
        # English
        if(data[i,j]=='E'){
          vecCOURSE<-c(1,0,0)
        }
      }
      # Spanish
    }
  }
```

```r
else if (data[i,j]=="S"){
  vecCOURSE<-c(0,1,0)
}
#German
else{
  vecCOURSE<-c(0,0,1)
}
#Job of specific purpose
if(j==2){
  #Health
  if (data[i,j]=="H"){
    vecJSP<-c(1,0,0,0)
  }
  #Tourism and Hospitality
  else if (data[i,j]=="T"){
    vecJSP<-c(0,1,0,0)
  }
  #Science and Technology
  else if (data[i,j]=="S"){
    vecJSP<-c(0,0,1,0)
  }
  #Business and Professional Language
  else{
    vecJSP<-c(0,0,0,1)
  }
  #Level
  if(j==3){
    #Level 1
    if(data[i,j]==1){
      vecLVL<-c(1,0)
    }
    #Level 2
    else{
      vecLVL<-c(0,1)
    }
  }
  #Overall Grade
  if (j==4){
    if (!is.na(data[i,j])){
      #high pass
      if (data[i,j]>0.90){
        vecG<-c(1,0,0)
      }
      #pass
      else if (data[i,j]<0.90 | data[i,j]>0.70){
        vecG<-c(0,1,0)
      }
      #low pass
      else if (data[i,j]<0.6){
        vecG<-c(0,0,1)
      }
    }
  }
  #Grammar Grade
  if (j==5){
    if (!is.na(data[i,j])){
      #high pass
      if (data[i,j]>0.90){
        vecGG<-c(1,0,0)
      }
      #pass
      else if (data[i,j]<0.90 | data[i,j]>0.70){
        vecGG<-c(0,1,0)
      }
      #low pass
      else if (data[i,j]<0.6){
        vecGG<-c(0,0,1)
      }
    }
  }
```
if (j==6){
    if (!is.na(data[i,j])){
        # high pass
        if (data[i,j]>0.90){
            vecGV<-c(1,0,0)
        }
        # pass
        else if (data[i,j]<0.90 | data[i,j]>0.70){
            vecGV<-c(0,1,0)
        }
        # low pass
        else if (data[i,j]<0.6){
            vecGV<-c(0,0,1)
        }
    }
}

# Speaking/Listening Grade
if (j==7){
    if (!is.na(data[i,j])){
        # high pass
        if (data[i,j]>0.90){
            vecGSL<-c(1,0,0)
        }
        # pass
        else if (data[i,j]<0.90 | data[i,j]>0.70){
            vecGSL<-c(0,1,0)
        }
        # low pass
        else if (data[i,j]<0.6){
            vecGSL<-c(0,0,1)
        }
    }
}

# Reading/Writing Grade
if (j==8){
    if (!is.na(data[i,j])){
        # high pass
        if (data[i,j]>0.90){
            vecGRW<-c(1,0,0)
        }
        # pass
        else if (data[i,j]<0.90 | data[i,j]>0.70){
            vecGRW<-c(0,1,0)
        }
        # low pass
        else if (data[i,j]<0.6){
            vecGRW<-c(0,0,1)
        }
    }
}

# Language of Specific Purpose Grade
if (j==9){
    if (is.na(data[i,j])){
        vecGLSP<-c(0,0,0,1)
    } else{
        # high pass
        if (data[i,j]>0.90){
            vecGLSP<-c(1,0,0,0)
        }
        # pass
        else if (data[i,j]<0.90 | data[i,j]>0.70){
            vecGLSP<-c(0,1,0,0)
        }
        # low pass
        else if (data[i,j]<0.6){
            vecGLSP<-c(0,0,1,0)
        }
    }
}
```r
# low pass
else if (data[i, j]<0.6){
  vecGLSP<-c(0,0,1,0)
}
}
vec<-c(vecCOURSE,vecJSP, vecLVL,vecG,vecGG,vecGV,vecGRW,vecGSL,vecGLSP)
mat<-rbind(mat,vec)
}
mat_n<-mat[2:nrow(mat),]
storage.mode(mat_n) <- "integer"
colnames(mat_n)<-c("E","S","G","H","TH","ST","BPL","L1","L2",
  "G_LP","G_P","G_HP",
  "G_G_LP","G_G_P","G_G_HP",
  "G_RW_LP","G_RW_P","G_RW_HP",
  "G_SL_LP","G_SL_P","G_SL_HP",
  "G_LSP_A","G_LSP_LP","G_LSP_P","G_LSP_HP")

trans <- as(mat_n, "transactions")

rules <- apriori(trans, parameter = list(supp = 0.01, conf = 0.8))
plot1<plot(rules, control=list(jitter=2))
## select and inspect rules with highest lift
rules_high_lift <- head(sort(rules, by="lift"), 5)
inspect(rules_high_lift)
## plot selected rules as graph
plot2<plot(rules_high_lift, method="graph", control=list(type="items"))
## graphs only work with very few rules
subrules2 <- sample(rules, 10)
plot3<plot(subrules2, method="graph")
plot4<plot(subrules2, method="graph",
  control=list(type="items"))
## grouped matrix plot 10% rules
subrules3<- sample(rules, (0.1*length(rules)))
plot5<plot(subrules3, method="grouped")

return(list(rules,plot1,plot2,plot3,plot5))
```

### Glossary

**E**: The learner learns English  

**S**: The learner learns Spanish  

**G**: The learner learns German  

**H**: The learner’s job of specific purpose is “Health”  

**TH**: The learner’s job of specific purpose is “Tourism & Hospitality”  

**ST**: The learner’s job of specific purpose is “Science & Technology”  

**BPL**: The learner’s job of specific purpose is “Business & Professional Language”
L1: The learner is at Level 1
L2: The learner is at Level 2
G_LP: The learner’s overall grade is “Low Pass”
G_P: The learner’s overall grade is “Pass”
G_HP: The learner’s overall grade is “High Pass”
G_G_LP: The learner’s overall grade at Grammar is “Low Pass”
G_G_P: The learner’s overall grade at Grammar is “Pass”
G_G_HP: The learner’s overall grade at Grammar is “High Pass”
G_V_LP: The learner’s overall grade at Vocabulary is “Low Pass”
G_V_P: The learner’s overall grade at Vocabulary is “Pass”
G_V_HP: The learner’s overall grade at Vocabulary is “High Pass”
G_RW_LP: The learner’s overall grade at Reading/Writing is “Low Pass”
G_RW_P: The learner’s overall grade at Reading/Writing is “Pass”
G_RW_HP: The learner’s overall grade at Reading/Writing is “High Pass”
G_SL_LP: The learner’s overall grade at Speaking/Listening is “Low Pass”
G_SL_P: The learner’s overall grade at Speaking/Listening is “Pass”
G_SL_HP: The learner’s overall grade at Speaking/Listening is “High Pass”
G_LSP_A: The learner’s overall grade at Language of Specific Purpose is “Absent”. (Because the
learner is at Level 1, and in this level the Language of Specific Purpose is not supported)
G_LSP_LP: The learner’s overall grade at Language of Specific Purpose is “Low Pass”
G_LSP_P: The learner’s overall grade at Language of Specific Purpose is “Pass”
G_LSP_HP: The learner’s overall grade at Language of Specific Purpose is “High Pass”