

# **Open University of Cyprus**

**Faculty of Pure and Applied Sciences**

## **Master's Thesis In Information Systems**



**Social Network Analysis Techniques and Text Mining to  
Assess the Learning Process of Students' Participation in the  
Online Discussion**

**Elvira Lotsari**

**Supervising Professor  
Vassilios Verykios**

**August 2014**

# **Open University of Cyprus**

**Faculty of Pure and Applied Sciences**

## **Social Network Analysis Techniques and Text Mining to Assess the Learning Process of Students' Participation in the Online Discussion**

**Elvira Lotsari**

**Supervising Professor**

**Vassilios Verykios**

The present thesis was submitted  
in partial fulfillment of the requirements

for a Master's degree  
in Information Systems

from the Faculty of Pure and Applied Sciences  
of Open University of Cyprus

**August 2014**

## Abstract

On a daily basis, a large amount of data is gathered through the participation of students in e-learning environments. This wealth of data is an invaluable asset to researchers as they can utilize it in order to generate conclusions and identify hidden patterns and trends by using big data analytics techniques. The purpose of this study is a threefold analysis of the data that are related to the participation of students in the online forums of their University. In one hand the content of the messages posted in these fora can be efficiently analyzed by text mining techniques. On the other hand, the network of students interacting through a forum can be adequately processed through social network analysis techniques. Still, the combined knowledge attained from both of the aforementioned techniques, can provide educators with practical and valuable information for the evaluation of the learning process, especially in a distance learning environment. In addition, we propose a classification via decision tree approach in order to forecast students' learning performance based on forum data. The study was conducted by using real data originating from the online forums of the Hellenic Open University (HOU). The analysis of the data has been accomplished by using the R tools, in order to analyze the structure and the content of the exchanged messages in these fora as well as to model the interaction of the students in the discussion threads.

**Keywords:** Data Analytics, Social Network Analysis, Text Mining, Educational Data Mining, Learning Analytics

# Acknowledgements

I would like to thank my supervisor professor Vassilios Verykios for his advice, guidance as well as his unwavering support throughout the dissertation.

I dedicate my thesis to my family  
for the support and assistance  
throughout the duration of my studies

# Contents pages

<b>1 Introduction</b> .....	1
1.1 Approach to the Problem .....	3
1.2 Structure of the Thesis .....	3
<b>2 Background-Definition</b> .....	5
2.1 Educational Data Mining.....	5
2.1.1 The Goals of EDM .....	6
2.2 Learning Analytics .....	7
2.2.1 The Goals of Learning Analytics.....	7
2.3 Intelligent Data Mining Techniques .....	8
2.3.1 Statistical Techniques.....	9
2.3.2 Classification (Decision Trees).....	9
2.3.3 Clustering (Hierarchical Clustering) .....	11
2.3.4 Text Mining .....	11
2.4 R Environment for Intelligent Data Analysis .....	15
2.4.1 Types of Objects in R .....	15
2.4.2 Basic Assignment and Operations .....	16
2.5 Packages in R .....	16
2.5.1 The Tm Package .....	17
2.5.2 The Rpart Package .....	18
2.5.3 The Igraph Package.....	18
<b>3 Related Work</b> .....	20
3.1 Predicting Student Performance.....	20
3.2 Social Network and Mining Techniques .....	22
<b>4 The Proposed Learning Analytics Methodology and the Experimental Results</b> ..	24

4.1 Description of the Data.....	25
4.2 Text Mining of Forum Data.....	25
4.2.1 Term Frequency .....	26
4.2.2 Word Cloud for Indicating the Content of Discussion Forum .....	27
4.2.3 Term Associations.....	28
4.2.4 Clustering Terms of Discussion Forum.....	29
4.2.5 Statistical Information for Student Participation in the Fora.....	30
4.3 Decision Tree for Predicting Students' Performance .....	34
4.4 Social Network Analysis of Forum Data .....	40
4.4.1 Network of Students.....	40
4.4.2 Network of Posts.....	<b>41</b>
4.4.3 Two-Mode Network of Students and Threads.....	43
4.4.4 Two-Mode Network of Terms and Posts.....	44
<b>5 Results and Future Work.....</b>	<b>46</b>
5.1 Evaluation and Results.....	46
5.2 Future Work.....	47
<b>6 Conclusion.....</b>	<b>49</b>
<b>Bibliography .....</b>	<b>50</b>
<b>A Installation Guidelines of R programming .....</b>	<b>A-1</b>
<b>B Functions in R.....</b>	<b>B-1</b>

# Chapter 1

## Introduction

Nowadays, the internet in conjunction with the progress of telecommunications technologies, have changed completely the way information and knowledge are transmitted and shared throughout the world. This fact has brought radical changes in education sector, especially in distance learning. The online fora have become one of the most popular communication tools in e-learning environments. That communication is one of the key factors of the learning procedure. For example, an online discussion forum provides to the students involved, motivation for collaboration and group-work for achieving a common goal with personal contribution from every participant. Moreover, an online forum is a significant source of information for educators, and this is why Learning Analytics and Educational Data Mining have become one of the most attractive fields among researchers. There is a growing number of courses delivered using e-learning environments, and computer-supported collaborative learning tools, such as Moodle (Modular Object-Oriented Dynamic Learning Environment), WebCT and Blackboard [26]. The online asynchronous discussions, which take place in these environments, play an important role in the students' collaborative learning. According to Brindley et al. [18], collaborative learning appears to increase the learner's sense of community, which is related to his satisfaction and retention. These factors are important for the students related to a distance education program, not only because they lead to their cognitive improvement but also because they prevent them

from dropping out. The students are actively engaged in sharing information and perspectives through the process of interaction with other students [28].

Moodle [14] is the most widespread learning platform in distance learning. It is an open source software designed to provide educators, administrators and learners with a single robust, secure and integrated system that create personalized learning environments with a focus on the interaction and the collaborative construction of content.

Blackboard [12] is a commercial e-learning suite that allows instructors to create e-learning courses and to develop custom learning paths for group or individual students, providing tools that facilitate the interaction, communication and collaboration between all actors.

A series of new trends, especially in education, that are changing the role of information in our lives and the way of how this information can be identified , that analyze and improve every activity we do, has resulted in the development of new environments, such as the Personal Learning Environments (PLE).

Personal Learning Environments (PLE) are systems that help learners take control of and manage their own learning [40]. Specifically these systems enable learners to:

- set their own learning goals
- manage their learning, both content and process
- communicate with others in the process of learning

Thus, students learn the required material by building and following their own learning maps.

Technically, the PLE represents the integration of a number of "Web 2.0" technologies like blogs, Wikis, RSS feeds, Twitter, Facebook, etc. - around the independent learner. It becomes, not an institutional or corporate application, but a personal learning center, where content is reused and remixed according to the student's own needs and interests. It becomes, indeed, not a single application, but a collection of interoperating applications an environment rather than a system [38].



## **1.1 Approach to the Problem**

In conventional teaching environments, educators are able to obtain feedback on student learning experiences in face-to-face interactions with students, enabling a continual evaluation of their teaching programs [2]. In order to choose the right learning procedure in a classroom the educator must observe the students' behavior, analyze historical data, and estimate the effectiveness of different pedagogical strategies. In distance education, respectively, this informal monitoring is not possible, due to the fact that e-learning environments lack a closer student-educator relationship. The main characteristic of this educational method is that the student is being taught and instructed without the physical presence of a tutor in a teaching classroom. For that reason, in these environments it is a significant challenge for the instructors to evaluate the participation of the students and analyze the structure of these interactions manually. Thus, educators must seek for other methods to obtain this information.

In this thesis, we use a Learning Analytics methodology and we discover information by linking patterns that are hidden in the educational contexts of students. Afterwards, we evaluate these patterns in order to improve the quality of the online student learning process at large. From the text messages exchanged among students, we extract useful information and we figure out certain points for providing personalized help [8]. In order to take full advantage of all this information derived from the participation of the students, we try to answer questions like "who is involved in each discussion?", or "who is the active/peripheral participant in a discussion thread?" [24].

For the above purpose, by using social network analysis techniques, we try to focus on the analysis of the interaction of students in online discussions. Text mining was conducted to explore patterns and trends through the content of the exchanged messages and finally classification via decision trees was implemented in order to predict students' final mark based on forum data. We used the well-known statistical software environment R [15] for the data analysis since it provides a broad range of statistical, data mining and visualization techniques.

## **1.2 Structure of the Thesis**

The rest of the thesis is structured as follows. Chapter 2 introduces some basic concepts and background of methods used for the implementation of this thesis while in Chapter 3, we outline the related work on Social Network Analysis and Mining on online forums with the main

emphasis to distinguish the contributions of this thesis. In Chapter 4, we describe the methodology that is followed for the analysis of our data and in Chapter 5, we justify its practicality by evaluating the experimental results produced. Finally, we conclude and summarize our findings in Chapter 6.

# Chapter 2

## Background-Definition

In this Chapter, we present the terminology and the background of all the methods that we used for the implementation of this thesis. Initially, we describe the educational data mining as well as learning analytics, two fields that are specific to the use of big data in education. Thereafter, we present the intelligent data mining techniques that were applied in order to uncover hidden and important patterns from students' interaction in the online forum aiming at improving the educational process. Finally, we present the statistical software environment R as well as the packages that were used in the embodiment of the dissertation.

### 2.1 Educational Data Mining

Educational Data Mining (EDM) is an emerging interdisciplinary research area that deals with the developments of methods that explore data originating from an educational context. More specifically, it applies different methods originating from statistics, machine learning and data mining in order to analyze data collected during the teaching and the learning process. Students'

learning data are being explored to develop predictive models and to discover new knowledge based on students' usage data. This procedure helps the educators evaluate educational systems, potentially improve some aspects of the quality of education and lay the groundwork for a more effective learning process [4].

### **2.1.1 The Goals of EDM**

Baker and Yacef [35] in their research suggest the following four goals of EDM:

*Predicting students' future learning behavior* – With the use of student modeling, this goal can be achieved by creating student models that incorporate the learner's characteristics, including detailed information such as their knowledge, their behaviours and their motivation to learn. The learner's experience and his overall satisfaction with learning are also measured.

*Discovering or improving domain models* – Through the various methods and applications of EDM, discovery of new and improvements to existing models are possible. Examples include the illustration of the educational content to engage learners and the determination of optimal instructional sequences to support the student's learning style.

*Studying the effects of educational support* that can be achieved through learning systems.

*Advancing scientific knowledge about learning and learners* by building and incorporating student models, the field of EDM research and the technology and software used.

In order to achieve the above goals, educational data mining researchers use five categories of technical methods which are *prediction, clustering, relationship mining, distillation of data for human judgment and discovery with models*.

The implementation of these methods, leads the researchers to create models in order to answer questions like:

- What sequence of topics is most effective for a specific student?
- What student actions are associated with more learning?
- What student actions indicate satisfaction, engagement, and learning progress?

- What features of an online learning environment lead to better learning?
- What will predict students success?

## 2.2 Learning Analytics

Learning analytics refers to the interpretation of a wide range of data produced by and gathered on behalf of students in order to assess academic progress, predict future performance, and spot potential issues. Data are collected from explicit student actions, such as completing assignments and taking exams, and from tacit actions, including online social interactions, extracurricular activities, posts on discussion forums, and other activities that are not directly assessed as part of the student's educational progress [23].

Unlike Educational Data Mining, Learning analytics does not examine the development of new algorithms or new models for data analysis but instead addresses the application of known methods and predictive models to answer important issues that affect student learning and instructional systems.

The methodology of learning analytics includes, (a) *the gathering of the data*, derived from the students and the learning environment in which they participate, and (b) *the intelligent analysis of this data* that leads to conclusions regarding the degree of the participation of the students in the forums and how this participation affects their learning.

### 2.2.1 The Goals of Learning Analytics

The main goal of Learning Analytics methodology is to understand and optimize the learning processes and also to improve the environments in which these processes occur [9].

Analytics have been used for:

- *Prediction purposes*, to identify, for instance, 'at risk' students in terms of drop out or course failure.
- *Personalization & adaptation*, to provide students with tailored learning pathways, or assessment materials.

- *Intervention purposes*, to provide educators with information to intervene and to support students.
- *Information visualization*, typically in the form of so-called learning dashboards which provide overview learning data through data visualization tools.

The implementation of the methodology of Learning Analytics, answers to the following questions:

- When are students ready to move on to the next topic?
- When are students failing behind in a course?
- When is a student at risk for not completing a course?
- What grade is a student likely to get without intervention?
- What is the best next course for a given student?
- Should a student be referred to a counselor for help?

## 2.3 Intelligent Data Mining Techniques

In this Section we present the intelligent data mining techniques that we will use in order to analyze our data. Specifically, in 2.3.1 we make use of statistical techniques such as *Term Frequency Analysis*, *Term Associations* and *Wordcloud*, that applied in order to find important terms that they reveal the content of discussion forum. In the next Subsection 2.3.2, we describe the Classification method via Decision Trees that we followed for monitoring and predicting students' learning performance while Subsection 2.3.3 gives the description of Hierarchical Clustering process with aiming to identify groups of terms with similar meanings. Thereafter, Subsection 2.3.4 refers to Text Mining Techniques to explore patterns and trends through the content of the exchanged messages while Subsection 2.3.5 presents the Social Network Analysis, that aims at studying relationships between students in online discussion.

### 2.3.1 Statistical Techniques

In this section we refer to the statistical techniques presented in thesis.

- *Term Frequency analysis*: It is a numerical statistic that is intended to reflect the importance of a word in a document in a collection or corpus.
- *Term Associations*: By using this method we can find the relationship between certain terms and how these terms appear in text source.
- *WordCloud*: It is a visual representation for text data that gives greater emphasis to terms that appear more frequently in the document.

### 2.3.2 Classification (Decision Trees)

Classification is a procedure that consists in predicting the value of a categorical attribute (the class) based on the values of other attributes (predicting attributes). A search algorithm is used to induce a classifier from a set of correctly classified data instances called the training set. Another set of correctly classified data instances, known as the testing set, is used to measure the quality of the classifier obtained [30]. Prediction of a student's performance is one of the oldest and most popular applications of Data Mining in education, and different kinds of models, such as Decision Trees or Rules, can be used to represent classifiers.

Decision trees (also referred to as classification and regression trees) are the traditional building blocks of data mining and the classic machine learning algorithm. Since their development in the 1980s, decision trees have been the most widely deployed machine-learning based data mining model builder. Their attraction lies in the simplicity of the resulting model, where a decision tree (at least one that is not too large) is quite easy to view, understand, and importantly, explain. Decision trees do not always deliver the best performance, and represent a trade-off between performance and simplicity of explanation. The decision tree structure can represent both classification and regression models [10].

## Decision Tree Methodology

The decision tree methodology can be summarized into three main steps: **a. Splitting Criterion**, **b. Pruning Procedure** and **c. Tree Selection** [31].

### *a. Splitting Criterion*

Splits are formed on subsets of the data in a *greedy* fashion. At the beginning, a variable and a split location is defined usually based on a particular criterion: Gini (classification), sums of squares (regression) from the entire dataset. The data is partitioned into two groups based on this split and the process is repeated on the two subgroups. Splitting continues until a large tree is constructed where only a small number of observations of the same class reside in each terminal node.

### *b. Pruning Procedure*

Pruning commences once a large tree has been grown. It involves successively snipping back splits of the tree into smaller trees using a cost complexity measure. This involves computing an error measure, usually calculated using cross-validation.

In detail, once a large tree has been grown, it is important to prune the tree back to avoid over fitting. This is important for two reasons:

- to ensure that the tree is small enough to avoid putting random variation into predictions
- to ensure that the tree is large enough to avoid putting systematic biases into predictions

Trees are pruned using a cost complexity measure which is defined as follows

$$R_{\alpha} = R + \alpha \times T$$

In the above expression, T represents the number of splits/terminal nodes in the tree, R represents the tree risk and  $\alpha$  represents the complexity parameter, which is a penalty term that controls the size of the tree. The estimate of tree risk differs depending on the type of tree



produced. For a classification tree, tree risk refers to the misclassification error, while for a regression tree, tree risk corresponds to the residual sum of squares

### *c. Tree Selection*

Tree selection is typically based on cross-validation and/or the use of a test dataset for larger applications. In addition, trees that are smaller in size but comparable in accuracy (corresponding to the *1 SE rule*) are also investigated.

A decision tree uses the following traditional structure. It starts with a single root node that splits into multiple branches, leading to further nodes, each of which may further split or else terminate as a leaf node. Associated with each nonleaf node will be a test or question that determines which branch to follow. The leaf nodes contain the "decisions" [10].

### **2.3.3 Clustering (Hierarchical Clustering)**

A hierarchical method creates a hierarchical decomposition of the given set of data objects forming a dendrogram - a tree which splits the database recursively into smaller subsets. The dendrogram can be formed in two ways: "bottom-up" or "top-down". In the "bottom-up" approach, also called as the "agglomerative" approach, initially each object forms a separate group. Then, the objects or the groups are successively merged according to some measures, like the distance between the two centers of two groups. This process continuous until all groups become one (the topmost level of the hierarchy), or until a termination condition holds. In the top-down approach, also called as the "divisive" approach, initially, every object is in the same cluster. In each subsequent iteration, a cluster is split into smaller clusters according to some measures until eventually each object results in one cluster, or until termination condition holds [19].

### **2.3.4 Text Mining**

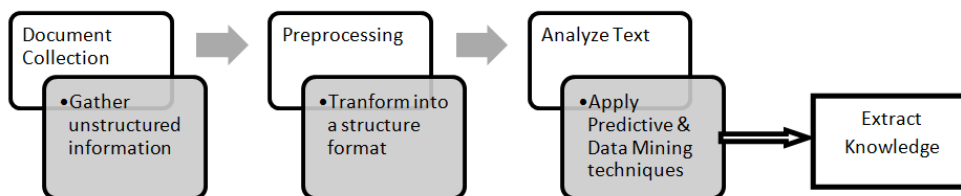
Text Mining refers to the process of deriving high-quality information from a text. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning. Text Mining usually involves the process of structuring the input text (usually parsing, along with the addition of some derived linguistic features and the removal of others, and subsequent insertion into a database), of deriving patterns within the structured

data, and finally of evaluating and interpreting of the output. 'High quality' in text mining usually refers to some combination of relevance, novelty, and interestingness [33].

Typical Text Mining tasks include: *text categorization, text clustering, information extraction, information retrieval, sentiment analysis, document summarization, and entity relation modeling* [20].

Text analysis is an interdisciplinary area comprising information retrieval, lexical analysis to study word frequency distributions, pattern recognition, tagging/annotation, information extraction, data mining techniques including link and association analysis, visualization, and predictive analytics. The main goal is, substantially, to turn text into data for analysis, via application of Natural Language Processing (NLP) and analytical methods [16]. Text Mining can function with unstructured or semi-structured datasets such as full-text documents, HTML files, emails, etc.

The Text Mining technique includes three individual processes which are: *Document Collection, Preprocessing, and finally Text Analysis* by implementation of predictive or Data Mining techniques. The following figure explores the detail processing methods in Text Mining.



**Figure 2.1:** Text Mining Process

As we can see in Figure 2.1, Text Mining starts with a collection of unstructured information and preprocesses it by checking format and character sets. Then it goes through a text analysis phase, sometimes by repeating techniques until information is extracted.

### 2.3.5 Social Network Analysis

The analysis of Social Networks (Social Network Analysis -SNA) views social relationships in terms of network theory, consisting of nodes (representing individual actors within the network)

and ties (which represents relationships between the individuals such as friendship, kinship, organizations, sexual relationships, etc.) [29][5]. These networks are often depicted in a social diagram, where nodes represent objects (actors) and the ties express relations.

## **Network Centrality**

Centrality is considered as one of the major and widely used conceptual tools for exploring actor roles in social networks. It refers to a group of metrics that aim to quantify the "importance" or "influence" (in a variety of senses) of a particular node (or group) within a network. There are three main measures of centrality: degree, betweenness, and closeness.

### **Degree Centrality**

Degree centrality is defined as the number of links incident upon a node (i.e., the number of ties that a node has). The degree can be interpreted in terms of the immediate risk of a node for catching whatever is flowing through the network (such as a virus, or some information). In the case of a directed network (where ties have direction), we usually define two separate measures of degree centrality, namely indegree and outdegree. Accordingly, indegree is a count of the number of ties directed to the node and outdegree is the number of ties that the node directs to others. When ties are associated to some positive aspects such as friendship or collaboration, indegree is often interpreted as a form of popularity, and outdegree as gregariousness [13].

We assume that the total number of actors in the network be  $n$ . The degree centrality [21] of an actor  $i$  ( $C_D(i)$ ) is the node degree (the number of edges) of the actor node, denoted by  $d(i)$ , normalized with the maximum degree,  $n-1$ . Thus, the Degree Centrality is given by the following formula:  $C_D(i) = \frac{d(i)}{n-1}$ .

The value of this measure ranges between 0-1 as  $n-1$  is the maximum value of  $d(i)$ .

### **Betweenness Centrality**

According to Borgatti [36], betweenness centrality, focuses on "the share of times that a node  $i$  needs a node  $k$  (whose centrality is being measured) in order to reach  $j$  via the shortest path". The more times a node lies on the shortest path between two other nodes, the more control the node has over the interaction between these two non-adjacent nodes [39].

The betweenness centrality of a vertex  $v$  in a graph  $G:=(V,E)$  with  $V$  vertices is computed as follows:

- For each pair of vertices  $(s,t)$ , compute the shortest paths between them.
- For each pair of vertices  $(s,t)$ , determine the fraction of shortest paths that pass through the vertex in question (here, vertex  $v$ ).
- Sum this fraction over all pairs of vertices  $(s,t)$ .

More compactly the betweenness centrality can be represented as: [43]

$$C_B(u) = \sum_{s \neq u \neq t \in V} \frac{\sigma_{st}(u)}{\sigma_{st}},$$

where  $\sigma_{st}$  is the total number of shortest paths from node  $s$  to node  $t$  and  $\sigma_{st}(u)$  is the number of those paths that pass through  $u$ . The betweenness may be normalized by dividing through the number of pairs of vertices not including  $v$ , which for directed graphs is  $(n-1)(n-2)$  and for undirected graphs is  $(n-1)(n-2)/2$ . For instance, in an undirected star graph, the center vertex (which is contained in every possible shortest path) would have a betweenness of  $(n-1)(n-2)/2$  (1, if normalized) while the leaves (which are contained in no shortest paths) would have a betweenness of 0.

### Closeness Centrality

A key node centrality measure in networks is closeness centrality [22], [39], [42]. Closeness centrality is based on the length of the average shortest path between a vertex and all vertices in the graph. Thus, we use the shortest distance to compute this measure. We assume that the shortest distance from path  $i$  to path  $j$  is  $d(i,j)$  [20]. The closeness centrality  $C_C(i)$  of path  $i$  is defined as

$$C_C(i) = \frac{n-1}{\sum_{j=1}^n d(i,j)}.$$

The value of this measure ranges between 0-1 as  $n-1$  is the minimum value of the denominator, which is the sum of the shortest distances from  $i$  to all other paths.

## 2.4 R Environment for Intelligent Data Analysis

R [34] is a free, cooperatively developed, open-source implementation of S, a powerful and flexible statistical programming language and computing environment that has become the effective standard among statisticians. It provides a wide variety of statistical and graphical techniques, including linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, and others. Furthermore, it compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. R is easily expandable through functions and extensions, and the R community is noted for its active contributions in terms of packages [32].

### 2.4.1 Types of Objects in R

R is an object oriented programming language. This means that virtually everything can be stored as an R object. Each object has a class. This class describes what the object contains and what each function does with it. There are several types of objects in R. The most important of them is Vector.

A brief description of the main types of objects in R follows:

- *Vectors*: A Vector is a sequence of data elements of the same basic type. Members in a vector are officially called components. Vectors are divided into six categories: logical, integer, real, complex, string (or character) and raw.
- *Arrays - Matrices*: An array can be considered as a multiply subscripted collection of data entries, for instance numeric. R allows simple facilities for creating and handling arrays, and in particular the special case of matrices. A matrix is a collection of data elements arranged in a two-dimensional rectangular layout.
- *Factors*: A factor is a vector object used to specify a discrete classification (grouping) of the components of other vectors of the same length. R provides both ordered and unordered factors.
- *Lists*: An R list is an object consisting of an ordered collection of objects known as its components. There is no particular need for the components to be of the same mode or type, and, for example, a list could consist of a numeric vector, a logical value, a matrix, a complex vector, a character array, a function, and so on.

- *Data Frames*: A data frame is a list of vectors, factors, and/or matrices all having the same length (number of rows in the case of matrices).
- *Functions*: Functions are themselves objects in R which can be stored in the project's workspace. They have three basic components: a formal argument list, a body and an environment.

## 2.4.2 Basic Assignment and Operations

R contains a number of operators listed in the Table 2.1 below:

**Table 2.1:** Basic Assignment and Operations

-	Minus, can be unary or binary
+	Plus, can be unary or binary
!	Unary not
~	Tilde, used for model formulae, can be either unary or binary
?	Help
:	Sequence, binary (in model formulae: interaction)
*	Multiplication, binary
/	Division, binary
^	Exponentiation, binary
%x%	Special binary operators, x can be replaced by any valid name
%%	Modulus, binary
%/%	Integer divide, binary
%*%	Matrix product, binary
%o%	Outer product, binary
%x%	Kronecker product, binary
%in%	Matching operator, binary (in model formulae: nesting)
<	Less than, binary
>	Greater than, binary
==	Equal to, binary
>=	Greater than or equal to, binary
<=	Less than or equal to, binary
&	And, binary, vectorized
&&	And, binary, not vectorized
	Or, binary, vectorized
	Or, binary, not vectorized
<-	Left assignment, binary
->	Right assignment, binary
\$	List subset, binary

## 2.5 Packages in R

Packages are collections of R functions, data and compiled code in a well-defined format which allow specialized statistical techniques, graphical devices (ggplot2), import/export capabilities, reporting tools (Knitr, Sweave), etc. These packages are developed primarily in R, and sometimes

in Java, C and FORTRAN. A core set of packages is included with the installation of R, with more than 5,800 additional packages and 120,000 functions (as of June 2014) available at the Comprehensive R Archive Network (CRAN), Bioconductor, and other repositories [11].

As it is already mentioned in the introduction of this chapter, in this section we refer to the most important packages that are used in this thesis such as: tm, rpart and igraph.

## 2.5.1 The Tm Package

Tm Package provides a framework for text mining applications within R. It offers functionality for managing text documents abstracts the process of document manipulation and eases the usage of heterogeneous text formats in R [17]. Table 2.2 gives an overview over the most-used methods for Text Mining provided by functions in package tm.

**Table 2.2:** Description of tm Package

Methods	Description
<b>Data Import</b>	<ul style="list-style-type: none"> <li>Specify the source to be character vectors.</li> <li>Create a data frame source</li> <li>Create a directory source</li> </ul>
<b>Inspect Objects</b>	Display detailed information with <code>tm_map</code> on a corpus or a term-document matrix.
<b>Transformations</b>	Predefined transformations which can be used: Converting to plain text documents, Eliminating extra whitespace, Convert to lower case, Remove stopwords, Stemming, Filters.
<b>Meta Data Management</b>	Accessing and modifying metadata of text documents and corpora.
<b>Creation of Term-Document Matrix</b>	Create a term-document matrix from a corpus.
<b>Operations on Term-Document Matrices</b>	<ul style="list-style-type: none"> <li>Accessing document IDs, terms, and their number of a term-document matrix or document-term matrix.</li> <li>Find associations in a document-term or term-document matrix.</li> <li>Find frequent terms in a document-term or term-document matrix.</li> <li>Read document-term matrices stored in special file formats.</li> </ul>
<b>Creation of Dictionary</b>	Creation of dictionary in order to restrict the dimension of the matrix and to focus on specific terms for distinct text mining contexts.

## 2.5.2 The Rpart Package

The rpart (Recursive Partitioning and Regression Trees) Package, is used to generate decision trees, to explore the structure of a set of data, while developing easy to reflect decision rules for predicting a categorical (classification tree) or continuous (regression tree) result. In Table 2.3 present an overview of the methods found in the rpart functions.

**Table 2.3:** Description of rpart Package

Methods	Description
<b>Building a Model</b>	To fit the decision tree
<b>Examine the Results</b>	<ul style="list-style-type: none"> <li>• depiction of cp(complexity parameter) table</li> <li>• plot cross-validation results</li> <li>• print results</li> <li>• detailed results including surrogate splits</li> <li>• plot decision tree</li> <li>• label the decision tree plot</li> <li>• creation of postscript plot of decision tree</li> </ul>
<b>Pruning the Tree</b>	Prune the tree back to the desired size to avoid overfitting the data.

## 2.5.3 The Igraph Package

"Igraph" is a library and R package for network analysis and visualization. The key objectives of the igraph library is to offer a set of data types and functions for a) pain-free implementation of graph algorithms, b) fast handling of large graphs, with millions of vertices and edges, c) allowing rapid prototyping via high level languages like R [11].

There are many functions that igraph package provides for creating graphs for social network analysis in R. In Table 2.4 presented some of the main functions used in the implementation of this thesis.



**Table 2.4:** Description of functions in igraph package

Functions	Description
<b>cliques()</b>	These functions find all, the largest or all the maximal cliques in an undirected graph
<b>degree()</b>	Degree and degree distribution of the vertices
<b>graph.adjacency()</b>	Is a flexible function for creating igraph graphs from adjacency matrices.
<b>graph.incidence()</b>	Creates a bipartite igraph graph from an incidence matrix.
<b>neighborhood()</b>	These functions find the vertices not farther than a given limit from another fixed vertex, these are called the neighborhood of the vertex.
<b>plotigraph()</b>	Is able to plot graphs to any R device.
<b>simplify()</b>	Simple graphs are graphs which do not contain loop and multiple edges
<b>tkplot()</b>	Interactive plotting of graphs

# Chapter 3

## Related Work

The goal of this section is to demonstrate how our work fits into the ecosystem of the already existing data mining and learning analytics techniques, as well as to indicate the novel elements of our unique approach.

### 3.1 Predicting Student Performance

Several types of classification methods and artificial intelligent algorithms that have been applied to predict student outcomes, marks or scores. Some of the more notable studies are presented below.

Kotsiantis et al. [41] have compared six classification methods (Naive Bayes, decision tree, feed-forward neural network, support vector machine, 3-nearest neighbor and logistic regression) to predict drop-outs in the middle of course. The data set contained demographic data, results of the first writing assignments and participation to group meetings. The data set contained 350

students. The best classifiers, Naive Bayes and neural network, were able to predict about 80% of drop-outs.

Namdeo et al. [44], in their study, applied four different classification algorithms for classifying students based on their final grade obtained in their courses. They used various attributes to build a predictive model, such as grades of homeworks during the semester. The algorithms that evaluated for prediction accuracy were, Decision Trees (ID3), Artificial Neural Network (multilayer perceptron), Naïve Bayesian Network and Decision Table classifier. After comparison and evaluation of algorithms, it was found that the best results given by the ID3.

Lopez et al. [25] describe the potential of the classification via a clustering approach in an educational context. The idea of using this kind of approach is to predict the final marks of the students by examining their participation in the forums. In this work, three experiments were carried out. Through these experiments the authors compared the accuracy of several clustering algorithms with that of traditional classification algorithms using the tenfold cross-validation method. The comparison was conducted in the base of predicting whether a student passes or not a course based on his participation in forums.

Minaei-Bidgoli et al. [1] present an approach to classifying students in order to predict their final grade based on features extracted from logged data in an education web-based system and then they demonstrate a genetic algorithm (GA) to successfully improve the accuracy of combined classifier performance, about 10-12% when comparing to non-GA classifier. Specifically, they have compared six classifiers (quadratic Bayesian classifier, 1-nearest neighbour, k-nearest neighbours, Parzen window, feed-forward neural network, and decision tree). The data contained attributes concerning each task solved and other actions like participating in the communication mechanism and reading support material. The data set contained 250 students. Experimental results were shown that the best classifier k-nearest neighbours, achieved over 80% accuracy, when the final results had only two classes (pass/fail). This method enables educators to identify students at risk, especially in classes with a large number of enrolled students, and allow them to help students to overcome the difficulties they encounter in a timely manner.

Osmanbegović et al. [7] have compared three data mining algorithms suitable for classification (Naive Bayes, Neural Networks and Decision Trees), in order to predict success in a course (either passed or failed) and the performance of the learning methods. The performances of the three models were evaluated based on the three criteria: the prediction accuracy, learning time and

error rate. The results indicate that the Naïve Bayes classifier outperforms in prediction decision tree and neural network methods since is both accurate and comprehensible for professors.

In [37], researchers apply data mining methodologies to study students' performance in the courses. For that purpose, they use for data classification task, the decision tree method. Information such as Attendance, Class test, Seminar and Assignment marks were collected from the student's management system, to predict the performance at the end of the semester. This research investigates the accuracy of Decision tree techniques for predicting student performance.

## **3.2 Social Network and Mining Techniques**

An interesting broad overview of recent studies on social network analysis techniques was presented in Rabbany et al. [26]. In their study, the authors described existing works and approaches on applying social network techniques for assessing the participation of the students in the online courses. They presented their specific social network analysis toolbox, named Meerkat-ED, for visualizing, monitoring and evaluating the participation of the students in a discussion forum. In particular, the visualization comprise the depiction of community detection among students on forum, keywords that indicate the topics addressed in the discussion and the relations between them, as well as, the centrality of students in the network. In addition, they present the implementation of Meerkat-Ed on their own case study data. Following this line of research, in our study, we use both text mining and social network analysis techniques to assess the learning process of students' participation in the online discussion.

Romero et al. [3] present a specific application of data mining in learning management systems and a case study tutorial with the Moodle system. In their study describe how different data mining techniques can be used in order to improve the course and the students' learning. They apply the most general and well known data mining techniques as well as two other specific data mining methods, the outlier analysis for data cleansing, spotting emerging trends and recognizing unusually good or bad performers, and the social network analysis for the analysis of the structure and context of online educational communities.

The work in [6] comprises part of the work accomplished in this thesis where a learning analytics methodology is proposed for student profiling. The authors use text mining and social network analysis techniques along with classification and clustering techniques, in order to draw conclusions and important patterns from raw data related to the participation of postgraduate students in the online forum of the module they have registered in. The study was conducted by using real data originating from the online forums of the Hellenic Open University (HOU). The analysis of the data has performed by using the R and the Weka tools, in order to analyze the structure and the content of the exchanged messages in these forums as well as to model the interaction of the students in the discussion threads. Following this line of research, in this dissertation, we tried to expand this study with more mining techniques, in order to achieve more accurately experimental results for assessing the learning process from students' participation in the online discussion.

# **Chapter 4**

## **The Proposed Learning Analytics Methodology and the Experimental Results**

In this Chapter, we demonstrate the methodology followed for the evaluation of the learning process of the students' online participation, by using real data from a postgraduate course offered in the context of distance education institute in Greece. For this purpose, in Section 4.1 there is a description of the data used for our experiments. In Section 4.2, we present the method of text mining in order to analyze the structure and the content of the exchange messages in the online forum, while in Section 4.3 we present the way of how to build decision trees with package rpart for monitoring and predicting student's learning performance. Following, in Section 4.4 we implement social network analysis techniques to uncover hidden and important patterns from the participation of the students in the online discussion.

## 4.1 Description of the Data

As we have already mentioned, the study was conducted by using real data originated from the Hellenic Open University (HOU), in Greek language. We draw this data from the Information Systems postgraduate program of study and, specifically, from the module named "Specialization in Software Technology". The data is anonymized by replacing the names of the students with the registration number (ID). We have built a dataset out of 64 students in the module, which consists of project (homework) performance, forum activity and final exam performance. By saying "homework" in a course module in the HOU, we mean six written assignments distributed during a 10-month period (spanning a full academic year). Forum activity includes the participation of each student, the messages exchanged between them, the discussion threads and finally, the student who started a topic.

## 4.2 Text Mining of Forum Data

Our analysis starts by extracting the text of messages from the discussion forum, with function `read.csv()`. Thereafter, the text is converted to a corpus which needs some modification such as removing of punctuations, numbers and hyperlinks. All these transformations are made by the `tm_map` function which implements a function to all elements of the corpus. After that, the corpus is transformed in order to build a document-term matrix. In this matrix, each row represents a term, each column represents a document and an entry in this matrix is the number of the occurrences of the term in the document.

The major problem encountered during the implementation of this thesis, concerned the word stemming which did not support the Greek language. In order to overcome this problem as well as to reduce the dimension of the matrix, we created a dictionary with specific terms pertaining to the learning materials of the module we studied. Thereby, we managed to present in the matrix only the words that included in dictionary. The dictionary can be created via the `Dictionary()` constructor. The Table 4.1 shows the terms that are contained in dictionary.

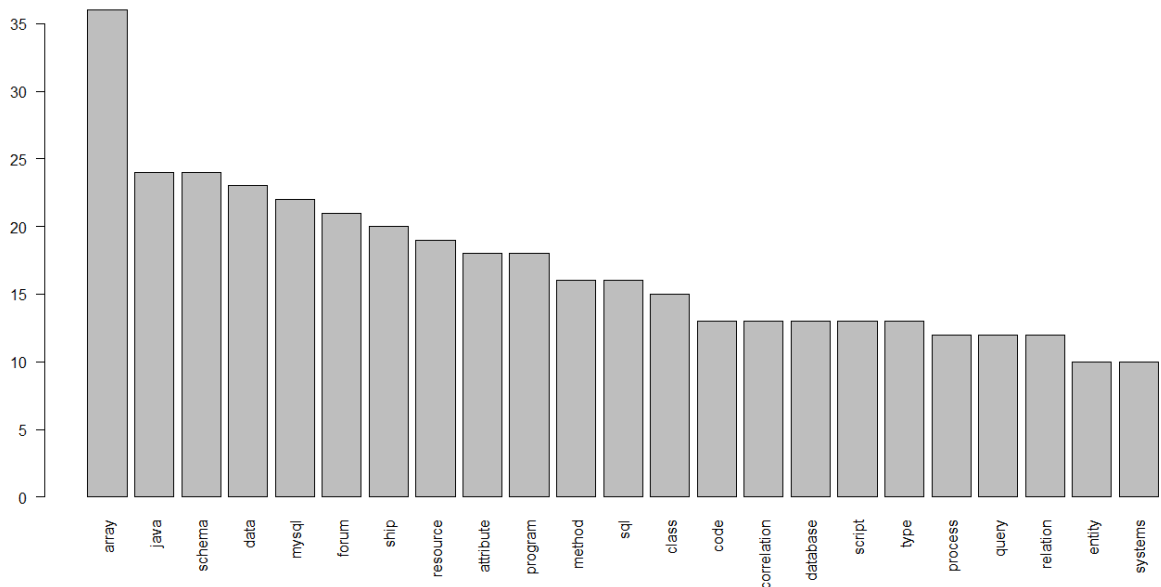
At this point, I would like to note that for reasons of homogeneity of presentation of the dissertation, we translated the Greek data in English. Thus, all results illustrated in charts, are in English language.

**Table 4.1:** Words of dictionary

<i>array</i>	<i>table</i>	<i>by</i>	<i>ubuntu</i>	<i>management</i>	<i>query</i>	<i>programming</i>
<i>bluej</i>	<i>eclipse</i>	<i>database</i>	<i>sequence</i>	<i>complicated</i>	<i>rdbms</i>	<i>code</i>
<i>constructor</i>	<i>page</i>	<i>having</i>	<i>server</i>	<i>submarine</i>	<i>constraint</i>	<i>method</i>
<i>exception</i>	<i>grep</i>	<i>mysql</i>	<i>objectoriented</i>	<i>data</i>	<i>relation</i>	<i>class</i>
<i>fire</i>	<i>bash</i>	<i>java</i>	<i>operating</i>	<i>type</i>	<i>er</i>	<i>battle</i>
<i>ide</i>	<i>shell</i>	<i>forum</i>	<i>systems</i>	<i>attribute</i>	<i>erd</i>	<i>slides</i>
<i>r</i>	<i>virtual</i>	<i>script</i>	<i>semaphore</i>	<i>ratio</i>	<i>diagram</i>	<i>os</i>
<i>resource</i>	<i>machine</i>	<i>sql</i>	<i>process</i>	<i>ternary</i>	<i>relational</i>	<i>pseudocode</i>
<i>segment</i>	<i>key</i>	<i>linux</i>	<i>program</i>	<i>weak</i>	<i>schema</i>	<i>tuples</i>
<i>ship</i>	<i>awk</i>	<i>from</i>	<i>operator</i>	<i>correlation</i>	<i>model</i>	<i>superclass</i>
<i>throws</i>	<i>sed</i>	<i>select</i>	<i>package</i>	<i>entity</i>	<i>algebra</i>	<i>object</i>

### 4.2.1 Term Frequency

In order to find links between words and groups of words, from the document-term matrix, we apply different data mining techniques. First, we find frequent terms with frequency greater or equal than ten with function `findFreqTerms()`. For visualizing the results we used package `ggplot2`. A plot of words along with their frequencies is presented in Figure 4.1.



**Figure 4.1:** A plot of terms along with their frequency of occurrence for terms that appear at least ten times in the forum discussions.



The barplot in Figure 4.1 clearly shows that the most popular words in the source text are "array", "java", "schema", "data", and "mysql". This probably indicates, that students more closely involved in topics related to previous concepts.

#### 4.2.2 Word Cloud for Indicating the Content of Discussion Forum

Another way to analyze the content of the exchanged messages in the fora is to create word cloud, whereby we can depict the significance of words. A word cloud can be a useful tool when we need to give emphasis to words that appear more frequently in the text source. It can be easily produced with package "wordcloud", which R tool provides. First we must transform the term-document matrix to a normal matrix, and subsequently calculate the frequency of terms. The most frequent words shown in the center of the cloud and indicate the main content of discussion forum.



**Figure 4.2:** Visualization of word frequency in discussion forum

As we observe in the above wordcloud in Figure 4.2, there are some set of frequent words which refer to specific discussion topics. For instance, "operating", "systems", "resource", "process", "script", "program", "linux", "shell", "virtual", "machine", belong in threads that focus in Operating

Systems. Other important words are "mysql", "sql", "array", "schema", "database", "attribute", "query", "key", "correlation", "model", "diagram", "entity", that indicate themes on Data Management. There are also some terms on the topic of Modern Programming Languages, as is clear from words "java", "class", "object", "blue j", "constructor", "ship", "submarine".

### 4.2.3 Term Associations

Another statistical technique for text mining is to find what is highly associated with a word or between a pair of words by using function "findAssocs" from tm package from R. For that purpose, we determine the term that we want to find associates for, as well as the lowest acceptable correlation limit with this term. The previous process returns a vector of words which are associated with predetermined word and correlation.

**Table 4.2:** Association between words

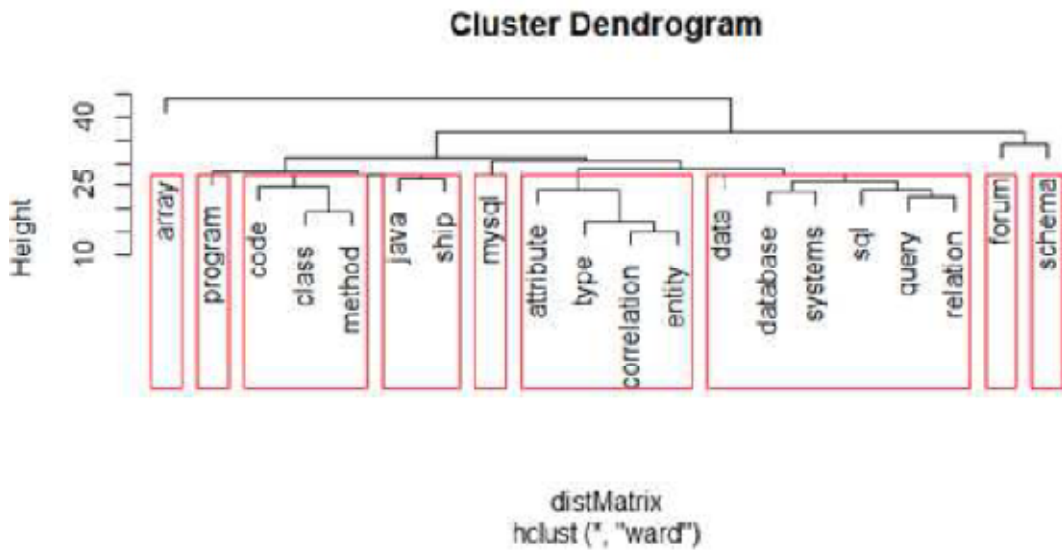
"Specific Word"	Words are Associated with "Specific Word"	Relationship between Words
"array"	tuples	0.32
	constraint	0.29
	submarine	0.27
	page	0.25
	table	0.25
"code"	battle	0.48
	superclass	0.27
"attribute"	complicated	0.58
	ternary	0.33
	correlation	0.29
"class"	<b>method</b>	<b>0.63</b>
	<b>superclass</b>	<b>0.46</b>
"correlation"	<b>entity</b>	<b>0.62</b>
	<b>type</b>	<b>0.47</b>
	<b>ternary</b>	<b>0.34</b>
	<b>attribute</b>	<b>0.29</b>
"data"	algebra	0.70
	management	0.57
	relational	0.49
	model	0.27
"virtual"	machine	1.00
	ubuntu	0.36
"entity"	type	0.63
	correlation	0.62
	erd	0.58
	package	0.28
"key"	relation	0.52
	schema	0.27

"mysql"	server sql algebra shell	0.56 0.34 0.31 0.26
"process"	resource semaphore ratio	0.98 0.52 0.38
"grep"	awk sed shell	1.00 1.00 0.4
"ternary"	correlation attribute complicated	0.34 0.33 0.28
"table"	page array	1.00 0.25
"resource"	process semaphore ratio	0.98 0.57 0.36
"schema"	diagram key complicated	0.48 0.27 0.25
"script"	sequence shell	0.26 0.26
"ship"	bluej submarine linux	0.49 0.31 0.29

Table 4.2 presents the reports related to the association between words, in descending order. For instance, the terms that associated with the word "correlation" with a value greater or equal than 0,25 are the following ones: "entity" (0.62), "type" (0.47), "ternary" (0.34), and "attribute" (0.29), which indicates a high relationship among terms related to one of the main subjects of the module, which is the Conceptual Database Modeling. Another example is the word "class" that is highly associated with the terms "method" (0.63) and "superclass" (0.46), all of them referring to the Java language.

#### 4.2.4 Clustering Terms of Discussion Forum

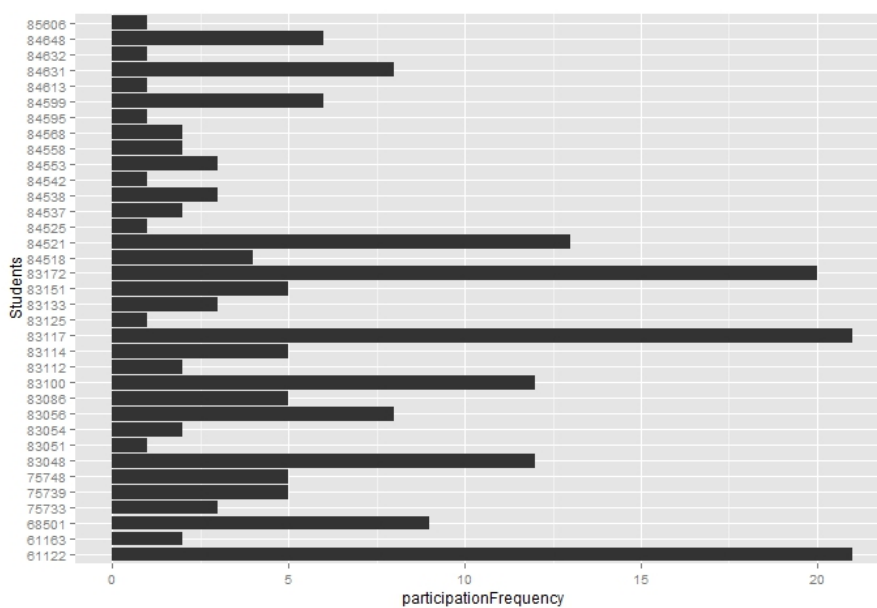
In our experiment, we applied hierarchical clustering, following an agglomerative method (bottom up). The heights (lengths of the lines within dendrogram) give indication of the level of correlation between terms, with shorter heights indicating stronger correlation. As we can see in Figure 4.3, the dendrogram is shaped into nine clusters. In each cluster, we can discern the discussion topic. For instance, the 3rd cluster, from the right, includes words such as "data", "database", "systems", "sql", "query" and "relation", which are all referring to the Relational Database Model.



**Figure 4.3:** Clusters of terms indicating frequently co-occurring terms in the discussion forum.

#### 4.2.5 Statistical Information for Student Participation in the Fora

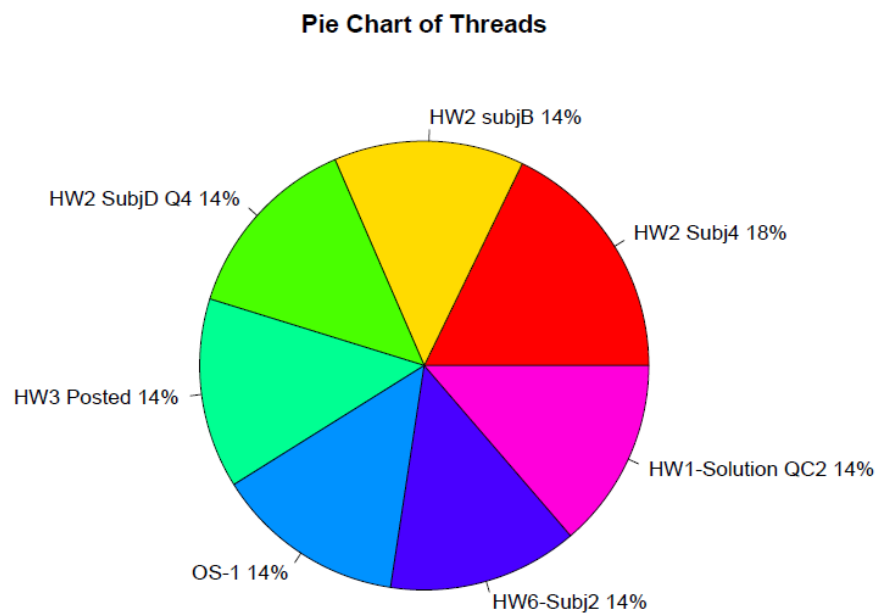
In Figure 4.4, we present the frequency of the participation of the students in the discussion forum. In the sequel, the threads that the students have mostly participated in are indicated in Figure 4.5. It is obvious that only a part of the population of students participated in the discussions by either initiating a new discussion thread or by replying to some other discussion started by another student.



**Figure 4.4:** Anonymized data for the participation of the students in the fora

According to Figure 4.4, the most "active" students are the ones with IDs 83117 and 61122. The frequency of their participation in the online forum is 21. Students with IDs 83172 and 84521 participate in 20 and 13 posts respectively follow in the most active students.

From the pie chart in Figure 4.5, we can observe that the most topics discussed by student, related with homework 2.



**Figure 4.5:** Pie Chart of the most frequent threads in the forum.

In order to select specific variables and observations from the data set, we use the `subset()` function. Thus, we select rows that have a value of participation no greater than 21. Then, we maintain the variables "student id", "start posting" and "final mark". The results are summarized in Table 4.3. The variable "start posting" represents the number of posts that each student has started, while "final mark" is the total score for each student.

**Table 4.3:** Indicators related to the participation of students in the forum along with final grade information for analyzing the effect of student participation.

STUDENT ID	PARTICIPATION	START POSTING	FINAL MARK
83117	21	2	6.9
61122	21	9	7.3
83172	20	7	9.0
84521	13	2	5.6
83100	12	5	6.8
83048	12	4	6.4
68501	9	1	-
84631	8	3	6.3
83056	8	4	8.0
84599	6	2	6.7
84648	6	3	7.2
83151	5	2	-
83114	5	-	8.4
83086	5	-	7.4
75748	5	1	7.1
75739	5	1	6.6
84518	4	-	7.1
84538	3	-	5.3
83133	3	2	5.4
75733	3	2	6.9
84553	3	-	7.7
84568	2	1	7.2
84558	2	1	6.8
84537	2	-	6.6
83112	2	-	-
83054	2	-	7.2
61163	2	1	-
85606	1	-	7.0
84632	1	1	8.7
84613	1	-	8.5
84595	1	1	6.2
84542	1	-	5.6
84525	1	-	7.2
83125	1	-	9.0
83051	1	-	6.1

We use the "table()" function that R provides, for creating frequency table for the discussion threads in the fora. The results appear in Table 4.4 in ascending order by frequency. In boldface are marked the most commonly discussion threads.

**Table 4.4:** Frequency table for Threads in ascending order

No. of Thread	Thread	Frequency
26th	HW2-Subj. 4th	13
19th	HW1-Solution of query C2	10
25th	HW2-Subj. B	10
31st	HW2-Subj. D -Q4	10
44th	Posting of HW3	10
68th	OS-1	10
88th	HW6-Subj. 2	10
48th	Many Happy Returns!	9
69th	HW5-Subj. 3B	9
67th	Grade of HW5 & Update Portal	8
79th	Hours of final examination on June	8
81st	Ubuntu-editor	8
9th	HW1-Sub. C1	7
20th	Solution of query C1	7
78th	HW5-Subj. 4-Reference Sequence	7
89th	Grades of previous homeworks	7
6th	HW1-Subj. C -Q2	6
8th	HW1-Sub. D	6
21st	Problem with the installation of MySQL	6
23rd	Question 26 pages 345	6
30th	HW2-Subj. D-Q6	6
37th	Management of date '9999-01-01' in field to_date in the table salaries	6
41st	HW2-Subj. B- Q2	6
61st	HW4-Subj. 3 (submarine)	6
74th	Shell Scripting	6
85th	Solution of HW5	6
11th	HW1-Sub. C	5
24th	Operator IN	5
27th	HW2-Subj. 4 query 1	5
33rd	HW2-Subj.A -Q5	5
38th	Clarification on subj. D4	5
50th	Passing the array to constructor	5
66th	HW5-Subj. 1	5
86th	Request for extension of HW6 Submission	5
3rd	Material of AGM1	4
5th	Curriculum	4
10th	Designation of building elements of ERD	4
29th	HW2-query for subj. D	4
64th	HW4-Orientation Ships	4
12th	File name for homework submission	3
18th	Version MySQL	3
32nd	HW2-Subj. D -Q5,6	3
39th	HW2-Subj. B -Q6	3
40th	HW2-Subj. A	3
42nd	e-Reader	3
43rd	Clarification on subj. A-Q4	3
46th	Step Over Compiler	3
49th	HW3-exercise 4	3
51st	HW3-Subj. 4	3
52nd	Questions at HW3	3
57th	Solution of HW3-Subj. 5	3
59th	HW4-Subj3-S1	3
60th	Study Program	3
62nd	HW4-Subj. 3 -S2	3
63rd	HW4-Numbering Shooter (available shots)	3
70th	Sourcelair	3
75th	Exercises for AGM3	3
82nd	HW5-Subj. 1-use of resources	3
2nd	Informing of the consultant instructor for absence from AGM	2
7th	Book	2
13th	HW1-Subj. D-Explanation	2
15th	Submission Form-Evaluation	2

16th	Deadline	2
22nd	HW2-Sub. A-Q2	2
28th	HW2-Subj. 4 query 1 (2)	2
34th	Today AGM2	2
45th	Solution of Subj. A1	2
47th	HW3-Subj. 4	2
53rd	HW3-3B2	2
54th	HW3-Subj. 4C	2
55th	Inheritance	2
58th	Solution of HW3-Subj. D1	2
65th	Stats Command	2
71st	Slides	2
72nd	HW5-Subject 1	2
73rd	HW5-Subj. 6	2
76th	OS-Material-2-virtual-mem-new.ppt	2
77th	HW5-Subj. 2-1	2
1st	Welcome to the new forum PLHS60	1
4th	Themes of final and iterative examinations	1
14th	Study Plan for the following weeks	1
17th	Posting of HW2 and HW1 Solution	1
35th	Material for AGM2	1
36th	Java Programming -Correspondences Study	1
56th	Posting of HW4	1
80th	Material of 4th AGM	1
83rd	Posting of HW6	1
84th	Submission of HW5	1
87th	Questionnaire promotion of your colleague	1

## 4.3 Decision Tree for Predicting Students' Performance

In this Section, we try to build predictive models with package rpart. The main goal is to forecast which students can successfully complete the modules as well as the final grade, based on seven known variables, which are the six homework grades (projects) and the participation in the online forum. The target variable is "total.mark", with the possible values of fail, good, very good and excellent. Before fitting the model, we first have to split data into two subsets: training (70%) and test (30%). The decision tree is built on the training data. In order to produce a really large tree, we set the complexity parameter (cp) very small, in value 0.000001.

There are two ways to represent the results of building a decision tree model: textual presentation using commands as "print()", "printcp()", "summary()" and plot that provides a better idea of what is a decision tree (Fig. 4.6). The outputs of the above commands are shown in the following Tables (4.5, 4.6 & 4.7).



In Table 4.5, we can see a textual version of the decision tree. The number of entities for each node is symbolized by character n, the loss denotes the number of entities that incorrectly classified, the yval is the default classification for the node and the yprobs is the distribution of classes in each node. A "\*" denotes that the tree is not split any further at that node. In any tree, the first node is the root node that is numbered as node number 1 and represents all observations. Therefore, in our case, the information contained tells us that the majority class of the root node is "very good", while the number 25 indicates how many of the 50 observations will be incorrectly classified.

**Table 4.5:** The output from print()

```
> print()

n= 50

node), split, n, loss, yval, (yprob)
 * denotes terminal node

1) root 50 25 very good (0.12000000 0.30000000 0.50000000 0.08000000)
2) proj2< 7.3 16 4 good (0.25000000 0.75000000 0.00000000 0.00000000)
4) proj3< 5.9 5 1 fail (0.80000000 0.20000000 0.00000000 0.00000000) *
5) proj3>=5.9 11 0 good (0.00000000 1.00000000 0.00000000 0.00000000) *
3) proj2>=7.3 34 9 very good (0.05882353 0.08823529 0.73529412 0.11764706)
6) proj3< 8.25 3 1 fail (0.66666667 0.33333333 0.00000000 0.00000000) *
7) proj3>=8.25 31 6 very good (0.00000000 0.06451613 0.80645161 0.12903226)
14) proj3< 9.45 10 2 very good (0.00000000 0.20000000 0.80000000 0.00000000) *
15) proj3>=9.45 21 4 very good (0.00000000 0.00000000 0.80952381 0.19047619)
30) proj5>=9.55 8 0 very good (0.00000000 0.00000000 1.00000000 0.00000000) *
31) proj5< 9.55 13 4 very good (0.00000000 0.00000000 0.69230769 0.30769231)
62) proj6< 9.05 9 1 very good (0.00000000 0.00000000 0.88888889 0.11111111) *
63) proj6>=9.05 4 1 excellent (0.00000000 0.00000000 0.25000000 0.75000000) *
```

Table 4.6 displayed the output of printcp() command that is useful in imaging the evolution of the CP values. The complexity parameter (CP) is used to control the size of the decision tree and to select an optimal tree size. For this to occur, we might choose to stop when the cross-validated error (xerror) begins to increase. In addition, the CP controls the process of pruning a decision tree. In Figure 4.7, we can see the graphical representation of the complexity table and to observe that the best tree is the tree with 4 terminal nodes corresponding to a complexity value of 0.046, since it is the tree that yields the lowest cross-validated error rate. Another piece of information that we can obtain for the above table is about variables that used in tree construction. Conceivably, only a subset of the variables will be used in the resulting decision tree model. Of the

seven input variables, only four are used in the tree construction which are proj2, proj3, proj5 and proj6.

**Table 4.6:** The output from printcp()

```
> printcp()

Classification tree:
rpart(formula = total.mark ~ proj1 + proj2 + proj3 + proj4 +
      proj5 + proj6 + participation, data = all.train, method = "class",
      parms = list(split = "information"), control = rpart.control(minsplit = 10))

Variables actually used in tree construction:
[1] proj2 proj3 proj5 proj6

Root node error: 25/50 = 0.5

n= 50

   CP nsplit rel error xerror  xstd
1 0.480000  0  1.00  1.00 0.14142
2 0.120000  1  0.52  0.76 0.13729
3 0.080000  2  0.40  0.68 0.13399
4 0.026667  3  0.32  0.56 0.12700
5 0.010000  6  0.24  0.60 0.12961
```

In Table 4.7 we can obtain information related to each node of the decision tree. Node 1 is the root node of the decision tree. We can see that it has 50 observations and a complexity parameter 0.48. The default class for this node in our case is "very good" which corresponds to the class that occurs most frequently in the training dataset. With this class as the decision associated with this node, the expected loss is 50% (0.5). The next line reports the frequency of observations by the target variable (total.mark). Thus, there are six observations with "fail" for "total.mark" (12%), fifteen with "good" (30%), twenty-five with "very good" (50%) and four with "excellent" (8%). The rest of the information relates to deciding how to split the node into two subsets. The resulting split has a left branch with sixteen observations and the right branch with thirty-four observations. The actual variable used to split the dataset into two subsets is proj2, with the test being on the value 7.3. The measure (the improvement), associated with the split of the dataset is 20.01.

**Table 4.7:** The output from summary()

```
< Summary()

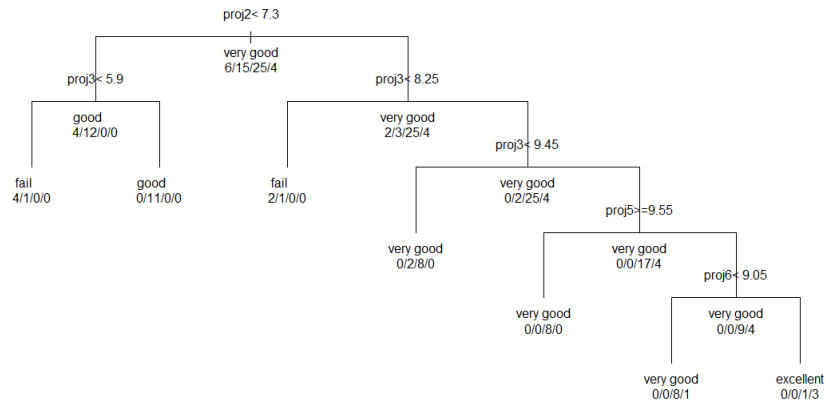
Node number 1: 50 observations,  complexity param=0.48
predicted class=very good  expected loss=0.5  P(node) =1
  class counts:  6  15  25  4
  probabilities: 0.120 0.300 0.500 0.080
  left son=2 (16 obs) right son=3 (34 obs)
  Primary splits:
    proj2 < 7.3  to the left,  improve=20.01835, (0 missing)
    proj3 < 8.25 to the left,  improve=19.28382, (0 missing)
    proj5 < 8.05 to the left,  improve=14.99024, (0 missing)
    proj4 < 7.2  to the left,  improve=11.03906, (0 missing)
    proj6 < 6.4  to the left,  improve= 9.96038, (0 missing)
  Surrogate splits:
    proj3 < 7   to the left,  agree=0.86, adj=0.562, (0 split)
    proj4 < 5.4 to the left,  agree=0.78, adj=0.312, (0 split)
    proj5 < 7.15 to the left, agree=0.78, adj=0.312, (0 split)
    proj6 < 4.85 to the left, agree=0.78, adj=0.312, (0 split)
    proj1 < 7.75 to the left, agree=0.76, adj=0.250, (0 split)
  ...

Node number 4: 5 observations
predicted class=fail  expected loss=0.2  P(node) =0.1
  class counts:  4  1  0  0
  probabilities: 0.800 0.200 0.000 0.000

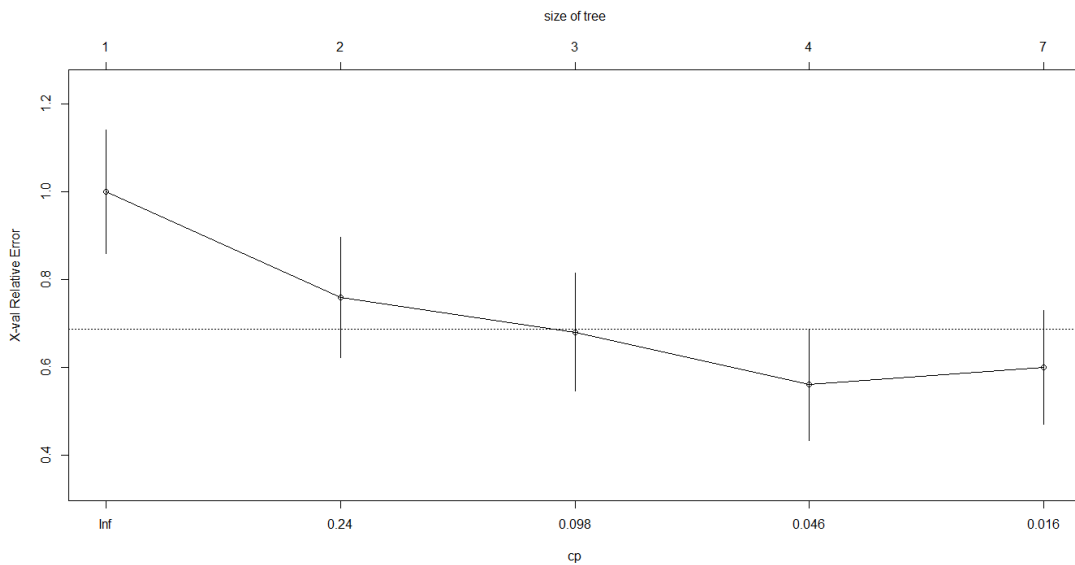
Node number 5: 11 observations
predicted class=good  expected loss=0  P(node) =0.22
  class counts:  0  11  0  0
  probabilities: 0.000 1.000 0.000 0.000
```

The surrogate splits that are next presented, related to the handling of missing values in the data. Consider the situation where we apply the model to new data but have an observation with proj2 missing. We could instead use proj3. The information here indicates that 86% of the observations in the split based on proj3<7. The adj value is an indication of what is gained by using this surrogate split over simply giving up at this node and assigning the majority decision of the new observation. Thus, in using proj3, we gain a 56.2% improvement by using the surrogate.

The other nodes listed in the summary, include the same kind of information. There are leaf nodes of the decision tree namely nodes 4, 5, 6, 14, 30, 62 & 63, will have the relevant information but no information on splits or surrogates. For instance, node 4 is a leaf node that predicts "fail" as the outcome. The expected loss is 20% and the probability of "fail" is 80%. Node 5 predicts "good" as the outcome, with expected loss 0% and the probability of good 100%.



**Figure 4.6:** Illustration of Decision Tree with package rpart (unpruned)



**Figure 4.7:** Graphical representation of the complexity table

### Selecting the Tree with Minimum Prediction Error

In order to select the tree with the minimum prediction error (Fig. 4.8), we use the information in the complexity table (Fig. 4.7) to prune the tree. As mentioned above, the best tree is the one that

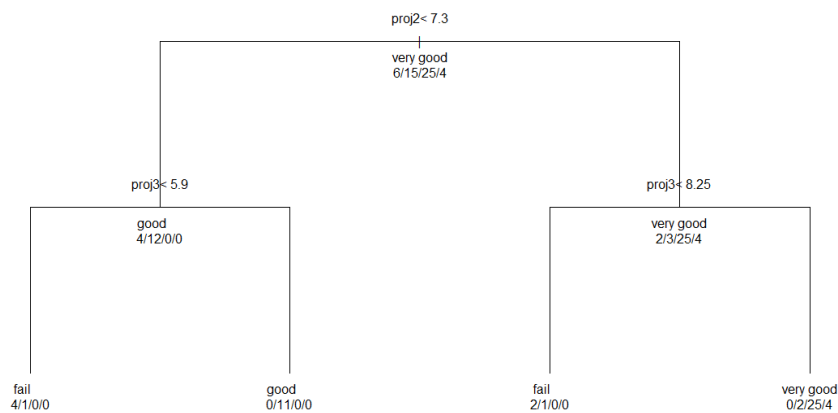
yields the lowest cross-validated error rate. Thus, we choose a complexity value greater than 0.046 but less than 0.098, which represents the tree with three splits.

The Figure 4.8 depicts the pruned tree which comprised of seven nodes and four leaf nodes, with maximum depth of 2.

The root is split into two nodes. The split is based on the variable proj2 with a split value 7.3. Node 2 has the split expressed as  $proj2 < 7.3$ . Only 4 of 16 observations are misclassified with this node being "good". This represents an accuracy of 75% in predicting that the final mark in the module will be "good".

The algorithm has chosen proj3 for the next split, with a split value 5.9. Node 4 has 5 observations for which the proj2 is less than 7.3 and the proj3 is less than 5.9. Under these conditions, the final mark in the module at 95%, will be "fail". Thereafter, node 5 has 11 observations and is clear that the final mark will be "good" at 100% when the proj2 is less than 7.3 and the proj3 is greater than or equal to 5.9.

Node 3 has 34 observations for which only 9 has misclassified with this node being "very good" on accuracy 73.5%. Next, follow node 6 with a 66.6% of probability final mark in the module will be "fail" when the proj2 is greater than or equal to 7.3 and the proj3 is less than 8.25. Finally, node 7 represents an accuracy of 80% in predicting that the final mark will be "very good" when the proj2 is greater than or equal to 7.3 and the  $proj3 \geq 8.25$ .



**Figure 4.8:** Illustration of Pruned Tree with package rpart

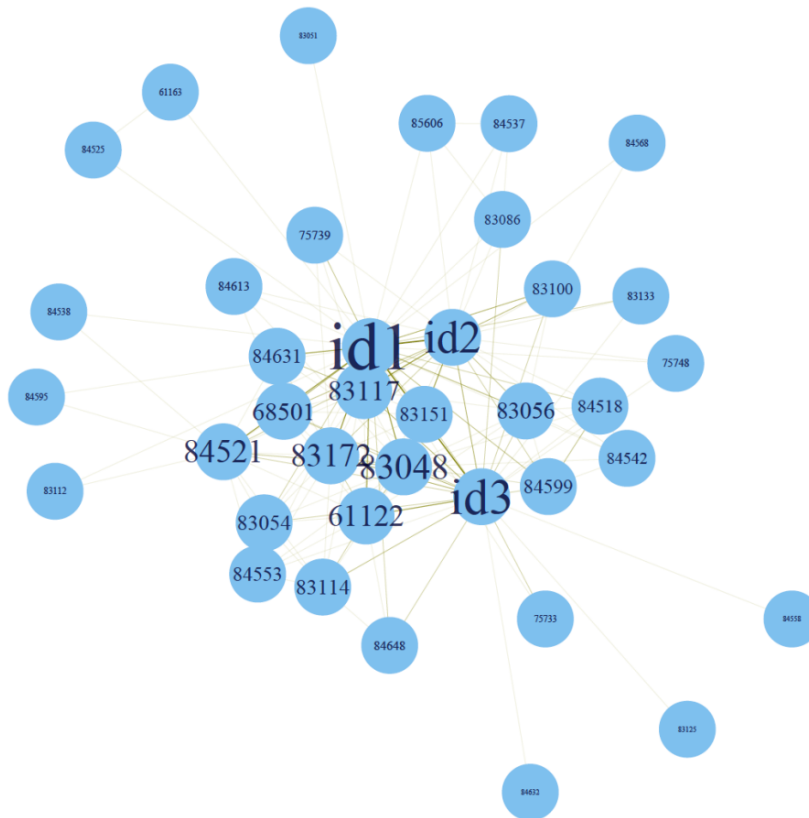
## 4.4 Social Network Analysis of Forum Data

In this section, we analyze the interaction of the students in the online forum as well as the correlation between the terms that are spotted in their discussions. Our analysis is conducted by using social network analysis techniques provided through various libraries in R. For that purpose, we built a network of students based on their co-occurrence in the same thread and a two-mode network that relies on both terms and posts.

### 4.4.1 Network of Students

First we built a network of students to illustrate the interactions among students in the same class. Each node represents a student and each edge represents a correlation between two students. The label size of vertices in the graph is based on their degree of participation and the width of edges is based on their weights. Thicker edges represent higher degree of correlation. In order to create the graph we used function `graph.adjacency()` from package `igraph`. The network of students is depicted in Figure 4.9.

From the graph, we can realize how influential a student is within this social network. The students with higher levels of participation in the discussion forum are at the center of the network. For instance, students with IDs 83117, 61122 and 83172 are located close to the center as, according to Table 4.3, they have the highest frequency of participation in the forum. In other words, they are the most active students in the class.

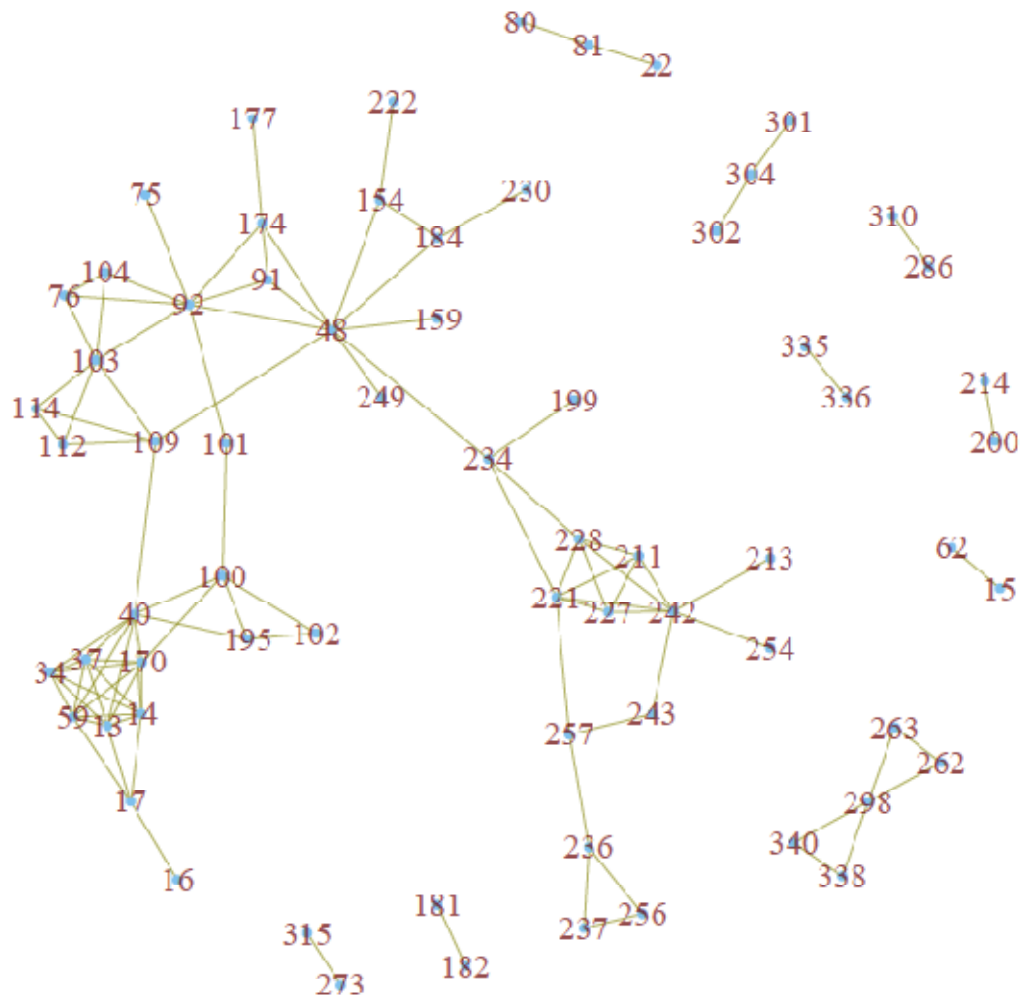


**Figure 4.9:** Network of students: The size of nodes indicates the degree of student's participation in the online forum as well as the centrality/leadership in the discussion. The labels id1, id2 and id3 are referring to course/module instructors.

#### 4.4.2 Network of Posts

In order to detect the discussion topics, we create a graph of threads based on the number of terms that they have in common. In this manner, we can find the relationship between the posts as well as the groups of them. To simplify the chart, we remove edges with low degree by using the function `delete.edges()`. This has the effect some vertices become isolated and are also removed. The generated graph is presented in Figure 4.10.

From the following chart in Figure 4.10, we can distinguish some cliques of posts which give the instructor a quick view of what is under discussion in this forum.



**Figure 4.10:** Network of Posts

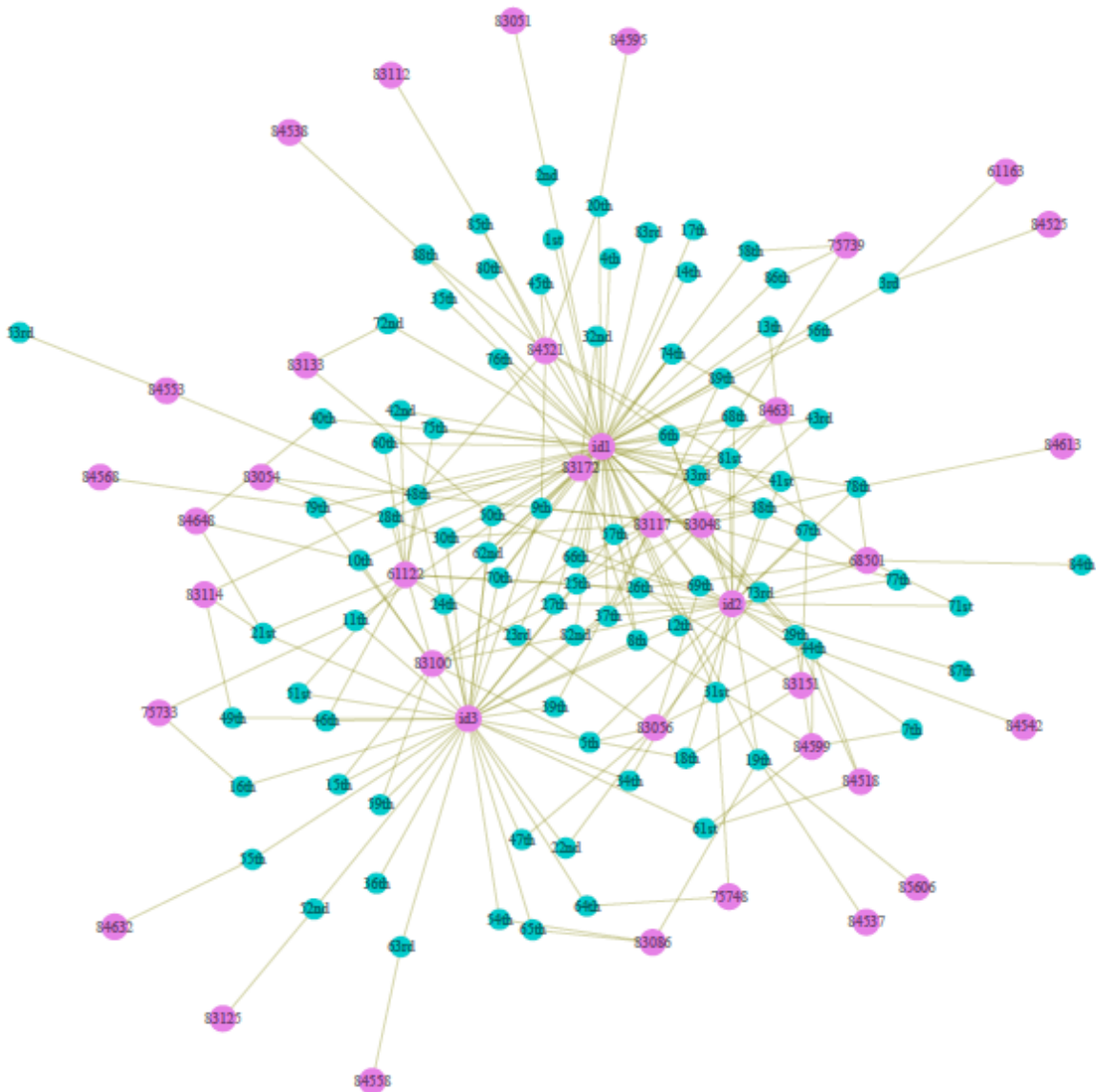
Using a specific code that R programming provides, we can demonstrate the posts that correspond to each post ID. For instance, the groups of posts 236, 237, 256, 257, 243, 242, 254 refer to the Homework 4 while posts 16, 17, 14, 13, 59, 34, 37 and 40 to the Homework 1. Other cliques of posts are shown below:

- Posts 76,75,92,91: Problem with installation of MySQL
- Posts 302,304,301: Shell Scripting
- Posts 338,340,298,262,263: HW5-Subj. 1-use of resources

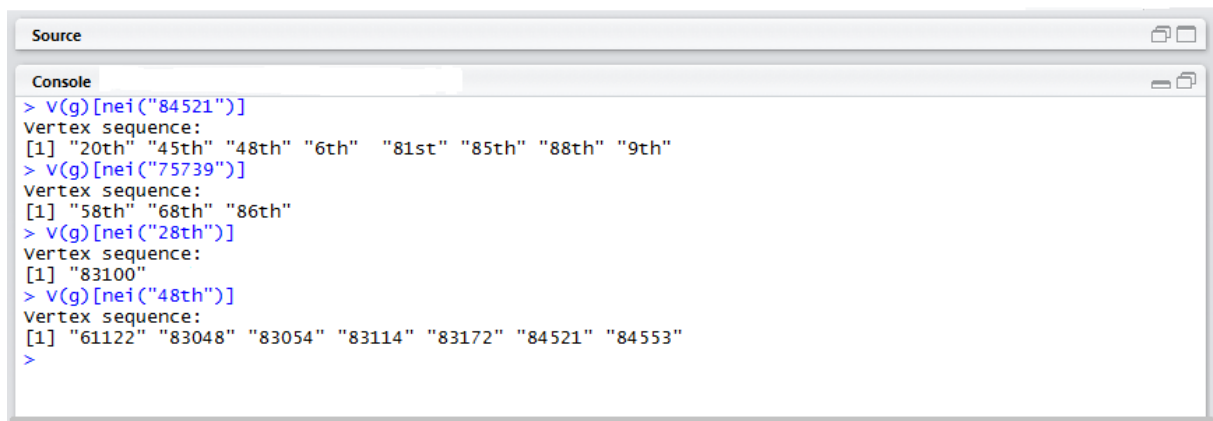


### 4.4.3 Two-Mode Network of Students and Threads

Subsequently, we built a two-mode network, which is composed of two types of nodes: students and threads with `graph.incidence()` which creates a bipartite graph from an incidence matrix. With the turquoise color, we illustrate the nodes of students and with purple color the nodes of threads. The graph is produced with a call to the `layout.fruchterman.reingold` function and illustrated in Figure 4.11, which represents the threads each student participates in, as well as how many different students are involved in each thread. For example, the student with ID 85606 participates in the 19th thread. However, in the 19th thread two more students are involved: 84537 and 85606.



**Figure 4.11:** A Two-Mode Network of Students and Threads



```
Source
Console
> v(g)[nei("84521")]
vertex sequence:
[1] "20th" "45th" "48th" "6th" "81st" "85th" "88th" "9th"
> v(g)[nei("75739")]
vertex sequence:
[1] "58th" "68th" "86th"
> v(g)[nei("28th")]
vertex sequence:
[1] "83100"
> v(g)[nei("48th")]
vertex sequence:
[1] "61122" "83048" "83054" "83114" "83172" "84521" "84553"
>
```

**Figure 4.12:** Snapshot of code, `nei(" ")`, that returns the threads that each student participates in, as well as how many different students are involved in each thread.

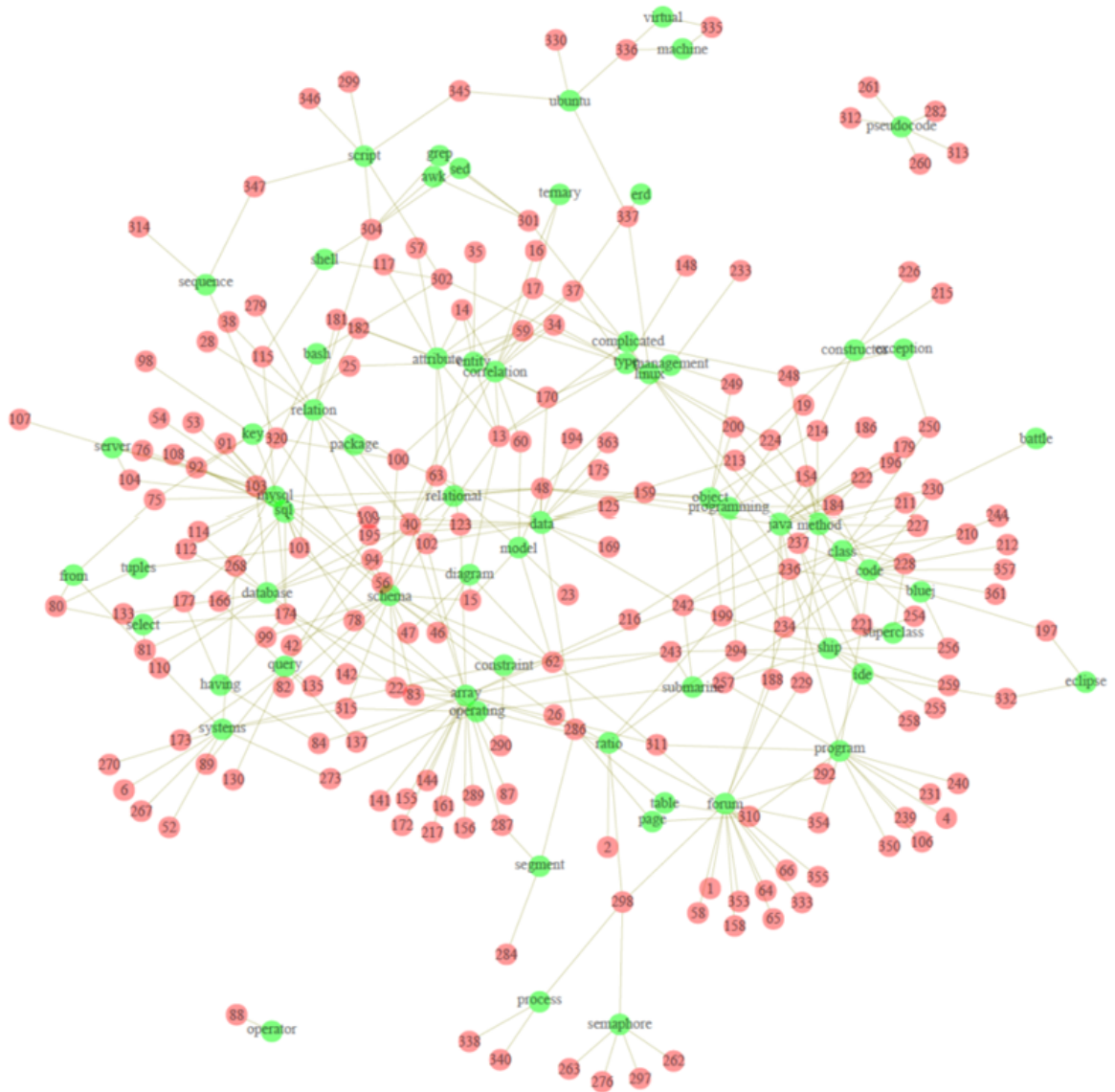
Another way to produce the above results is by using the code `nei(" ")`. A snapshot of code is presented in Figure 4.12.

#### 4.4.4 Two-Mode Network of Terms and Posts

Finally, we built a two-mode network which consists of both terms and posts. With green color we demonstrate the vertices of terms and with pink color the vertices of posts.

Figure 4.13 presents a group of posts and their associated keywords, such as "operating systems", "mysql database management system", "page table", "virtual machine", "sql query", "programming method", "key constraint", "BlueJ Java IDE", "linux shell script", "awk-sed-grep", and "ternary correlation".

Through the R statistical package, we can select the discussions that contain a specific group of terms we are interested in. Thus, we can distinguish the topics in which the majority of students are involved, and as result, we can identify either any weaknesses a group of students possibly has with respect to the understanding of the study materials or areas where certain study materials are not adequately explained for the whole class.



**Figure 4.13:** A Two-Mode Network of Terms and Posts

# Chapter 5

## Results and Future Work

In this Chapter, we attempt to assess the experimental results produced by the application of the intelligent data mining techniques that we followed for our data analysis. Finally, in Section 5.2 we present our future plan.

### 5.1 Evaluation and Results

In this work we used traditional data mining, along with text mining and social network analysis techniques in order to analyze data originating from the participation of students in discussion forums along with data related to the performance of students in the module. We managed, through graphs, to outline the profile of the students who participate in a discussion online forum. Specifically, Figure 4.4 and Figure 4.9 give us the frequency of participation of students and they demonstrate the interaction between them. Therefore, an instructor is provided with better means to evaluate participation in the online forum and to distinguish the active as well as the peripheral students.

From Figure 4.11, an instructor can derive information about the students that participate in specific threads and also how many different students are involved in each thread. Thereby, s/he has an overall view of the difficulties that the students in the module may face. In addition, from Figure 4.13, we can discern the topics in which the majority of students are involved. This enables the instructor to focus his attention on some specific concepts in his course. By doing so, he will try to enrich his educational material and he will ameliorate the learning process. The topics that are mostly discussed on the discussion forum stand out in Figure 4.5 while frequency table of threads is presented in Table 4.4.

By applying statistical and visualization techniques, to analyze the content of the exchanged messages in the fora, we achieved to discern important terms which reveal the discussion topics from the interaction between students in the course. Thereby, it gives the tutor a quick view of what is under discussion in this forum. These results appear in Figures 4.1, 4.2, 4.3, 4.10 and Table 4.2.

The experimental results in Section 4.3 give us the ability to forecast students' behavior and performance, based on seven known variables, as well as to evaluate the relevance of the attributes involved. Prediction technique helps educators to detect students at risk of failing the module and also to discover relationships between students' knowledge levels. Therefore, instructors can understand the e-learners needs, and they are able to provide them with scalable feedback and learning recommendations. This contributes to the improvement of virtual courses and learning process in general.

At the end, and by looking at the characteristics of each student demonstrated in Table 4.3, we can deduce that their final mark is not strongly related to their participation in the discussion forum. We could say that the performance of the mediocre students, regarding their courses is improved by expressing their queries and by exchanging messages and views with their fellow students and also their instructors.

## **5.2 Future Work**

In the future, we plan to analyze a dataset of bigger volume and variability that it will consist of data about postgraduate students that have a temporal dimension, observing the progress students make as they move along the thematic modules of the entire program of study.

Another interesting field that we would like to address is Sentiment Analysis. It is the best way to analyze postings of students and classify them in different types of emotions, as positive or negative. Thus, the sentiment analysis techniques can identify the students' positive or negative feelings and as in return it can help the instructors understand the students learning behaviour.

# Chapter 6

## Conclusion

In this study, we present a learning analytics methodology that we followed for student profiling. We used text mining and social network analysis techniques along with classification and clustering techniques, in order to draw conclusions and unearth important patterns from raw data related to the participation of postgraduate students in the online forum of the module they have registered in. We use R software environment for the data analysis and mining in order to illustrate who is involved in each discussion and who is the active/peripheral participant in a discussion thread. In addition, we manage to visualize the groups of terms that were mostly discussed in the online forum. Moreover, we summarize the characteristics of each student in a table, from which we can conclude that students' final mark is not based on their participation on the discussion forum. Other complementary results generated with classification and clustering techniques are also enlightening about the complicated process of the student learning in a group of peers and from a distance.

## Bibliography

- [01] B. Minaei-bidgoli, D.A. Kashy, G. Kortmeyer, W.F. Punch (2003). Predicting student performance: an application of data mining methods with an educational Web-based system LON-CAPA, in: Proceedings of the ASEE/IEEE International Conference on Frontiers in Education, Boulder, 2003, pp. 13–18.
- [02] C. Romero, & S. Ventura (2007). Educational data mining: A survey from 1995 to 2005. *Expert systems with applications*, 33(1), 135-146.
- [03] C. Romero, S. Ventura, & E. García (2008). Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 51(1), 368-384.
- [04] C. Romero, S. Ventura, (2010). Educational data mining: a review of the state of the art. *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, IEEE Transactions on, 40(6), 601-618.
- [05] D'Andrea, Alessia et al. (2009). "An Overview of Methods for Virtual Social Network Analysis". In Abraham, Ajith et al. *Computational Social Network Analysis: Trends, Tools and Research Advances*. Springer. p. 8. ISBN 978-1-84882-228-3.
- [06] E. Lotsari, V. Verykios, C. Panagiotakopoulos, & D. Kalles (2014). A Learning Analytics Methodology for Student Profiling. In *Artificial Intelligence: Methods and Applications* (pp. 300-312). Springer International Publishing.
- [07] E. Osmanbegović, & M. Suljić (2012). Data mining approach for predicting student performance. *Economic Review*, 10(1).
- [08] F. Abel, I. Bittencourt, E. Costa, N. Henze,, D. Krause, & J. Vassilev (2010). Recommendations in Online Discussion Forums for E-Learning Systems. *IEEE Transactions on Learning Technologies*, 3, 2, 165-176, 2010.
- [09] G. Siemens and R. S. J. Baker (2012). "Learning Analytics and Educational Data Mining : Towards Communication and Collaboration," in LAK12, 2012.



- [10] G. Williams (2009). Rattle: a data mining GUI for R. *The R Journal*, 1(2), 45-55.
- [11] <http://cran.r-project.org/web/packages/igraph/index.html>
- [12] [http://en.wikipedia.org/wiki/Blackboard\\_Learning\\_System](http://en.wikipedia.org/wiki/Blackboard_Learning_System)
- [13] <http://en.wikipedia.org/wiki/Centrality>
- [14] <http://en.wikipedia.org/wiki/Moodle>
- [15] [http://en.wikipedia.org/wiki/R\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/R_(programming_language))
- [16] [http://en.wikipedia.org/wiki/Text\\_mining](http://en.wikipedia.org/wiki/Text_mining)
- [17] I. Feinerer (2014). "Introduction to the tm Package Text Mining in R." nd): n. pag. Web (2014).
- [18] J. E Brindley, C. Walti, L. M Blaschke (2009). Creating Effective Collaborative Learning Groups in an Online Environment. *IRRODL*, 10, 3, 2009.
- [19] J.Han, M. Kamber (2000). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.
- [20] K. Vidhya. & G. Aghila (2010). *Text Mining Process, Techniques and Tools: an Overview* July 2010
- [21] L. Bing (2011). *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Springer. p. 271. ISBN 978-3-642-19459-7.
- [22] L. C. Freeman, 1978. Centrality in social networks: Conceptual clarification. *Social Networks* 1, 215-239.
- [23] L. Johnson, et. al (2011). *The 2011 horizon report*.
- [24] M. de Laat, V. Lally, L. Lipponen, & R.-J (2007). *Simons. Investigating patterns of interaction in networked learning and computer-supported collaborative learning: A role for social*

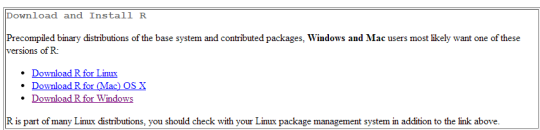
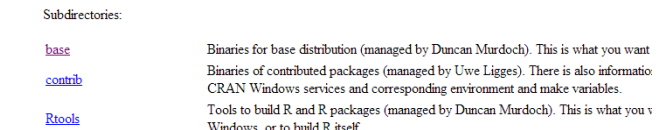
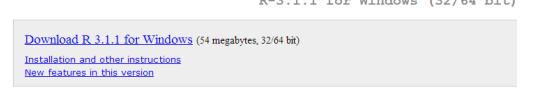
- net-work analysis. *International Journal of Computer-Supported Collaborative Learning*, 2(1):87-103, March 2007.
- [25] M. I. Lopez, et al. (2012) "Classification via clustering for predicting final marks based on student participation in forums." *Educational Data Mining Proceedings, 2012*".
- [26] M. Takaffoli, & O. R. Zaïane (2012). Social network analysis and mining to support the assessment of on-line student participation. *ACM SIGKDD Explorations Newsletter*, 13(2), 20-29.
- [27] Mining, Through Educational Data (2012). "Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics: An Issue Brief." (2012).
- [28] N. Y. Erlin, and A. A. Rahman. Students' interactions in online asynchronous discussion forum: A social network analysis. *Education Technology and Computer, International Conference on*, 0:25-29, 2009.
- [29] P. A. R. Carlos A.R. (2011). *Social Network Analysis in Telecommunications*. John Wiley & Sons. p. 4. ISBN 978-1-118-01094-5.
- [30] P. G Espejo, S. Ventura, & F. Herrera (2010). A survey on the application of genetic programming to classification. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(2), 121-144.
- [31] P. Kuhnert, B. Venables, & S. S. Zocchi, (2005). *An introduction to R: software for statistical modelling & computing*.
- [32] R Development Core Team, 2012. R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
- [33] R. Garg. "Study of text based mining (2011)." *Proceedings of the International Conference on Advances in Computing and Artificial Intelligence*. ACM, 2011.
- [34] R. Ihaka, and R. Gentleman. (1996). "R: A language for data analysis and graphics." *Journal of Computational and Graphical Statistics* 5:299-314

- [35] R.S. Baker, & K. Yacef (2009). The state of educational data mining in 2009: A review and future visions. *JEDM-Journal of Educational Data Mining*, 1(1), 3-17.
- [36] S. P. Borgatti, (2005). Centrality and network flow. *Social Networks*, 27(1), 55-71
- [37] S. Pal, & B. K. Baradwaj (2012). Mining educational data to analyze students' performance. arXiv preprint arXiv:1201.3417.
- [38] S. Schaffert. & W. Hilzensauer (2008). On the way towards Personal Learning Environments: Seven crucial aspects. In: *e-learning Papers*, 9 (2008)
- [39] S. Wasserman, & K. Faust (1994). *Social network analysis: Methods and applications*. New York: Cambridge university press.
- [40] S. Wilson, et al. (2006). Personal Learning Environments: Challenging the dominant design of educational systems, *Journal of e-Learning and Knowledge Society* |ISSN (online) 1971 - 8829 | ISSN (paper) 1826 - 6223
- [41] S.B. Kotsiantis, C.J. Pierrakeas, and P.E. Pintelas (2003). Preventing student dropout in distance learning using machine learning techniques. In *Proceedings of 7th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES)*, pages 267–274.
- [42] T. Opsahl, F. Agneessens, J. Skvoretz (2010). "Node centrality in weighted networks: Generalizing degree and shortest paths". *Social Networks* 32 (3): 245-251
- [43] U. Brandes, (2001). "A faster algorithm for betweenness centrality" (PDF). *Journal of Mathematical Sociology* 25: 163–177. doi:10.1080/0022250x.2001.9990249. Retrieved October 11, 2011.
- [44] V. Namdeo et. al (2010). Result Analysis Using Classification Techniques. 2010 *International Journal of Computer Applications* (0975-8887), 1(22

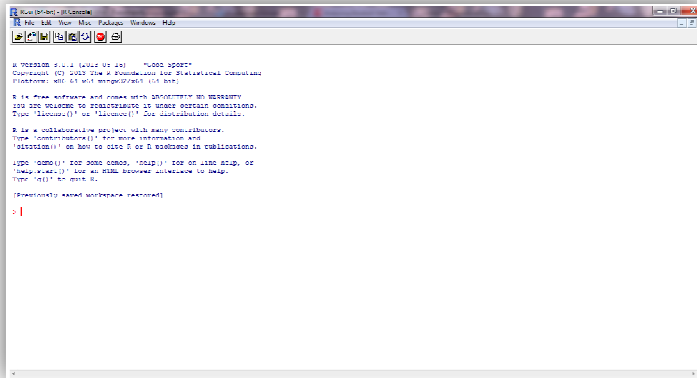
# Appendix A

## Installation Guidelines of R programming

The application is implemented entirely in R 3.1.1 and has been tested in this version, in Windows environment .Different versions of R may give incorrect results or give some kind of error message. Therefore, the first step that needed in order to use someone the tool, is to install on computer the R 3.1.1, which can be done easily via the link <http://cran.r-project.org/> and follow the steps below:

 <p>1. Select "Download R for Windows"</p>	 <p>2. Select "base"</p>
 <p>3. Select "Download R 3.1.1 for Windows"</p>	<p>4. The R Setup Wizard will appear in a window.</p> <p>5. Select "Next" at the bottom of the R Setup wizard window.</p> <p>6. R should now be installed. This will take about a minute. When R has finished, you will see "Completing the R for Windows Setup Wizard" appear. Select "Finish"</p> <p>7. To start R, you should click on the "R" icon.</p>

## 8. The R console should appear:



```

R Console
File Edit View Misc Packages Windows Help

R logo
R: the software and data with 2000+ users in 100+ countries
and 100+ journals to reproduce it using standard notations.
Type "license()" or "license()" for distribution details.
R is a collaborative project with many contributors.
Type "contributors()" for more information and
"citation()" on how to cite R or R packages in publications.
%%>> "demo()" for some demos, "help()" for on-line help, or
"help.wanted()" for an HTML browser interface to help.
Type "??()" to start R.
#install.packages("base")

> |
```

ImageA.1: The R GUI

# Appendix B

## Functions in R

This annex, provides the basic functions that used for implementation of this thesis.

Function	Description
<code>abline()</code>	Plots a straight line to an existing graph
<code>as.matrix()</code>	Attempts to turn its argument into a matrix
<code>as.numeric()</code>	Convert another data type to a number
<code>attach()</code>	Makes the variables of a dataset available without \$
<code>barplot()</code>	Creates a bar chart

<b>c()</b>	Combines its arguments
<b>coord_flip()</b>	Swaps x and y axis
<b>cut()</b>	Divides a numeric vector into different ranges
<b>data.frame()</b>	Makes a data frame from separate vectors
<b>detach()</b>	Quash an attach function
<b>dev.off()</b>	Closes files
<b>diff()</b>	Returns suitably lagged and iterated differences
<b>dim()</b>	Gets or sets the dimension of a matrix, array or data frame
<b>dist()</b>	Computes and returns the distances between the rows of a data matrix
<b>do.call()</b>	Calls a function with a variable number of arguments
<b>E()</b>	Access to the edges of graphs
<b>findAssocs()</b>	Finds associations in a document-term or term-document
<b>findFreqTerms()</b>	Finds frequent terms in a document-term or term-document matrix
<b>graph.adjacency()</b>	Builds igraph graphs from adjacency matrices
<b>grep()</b>	Pattern matching

<b>gsub()</b>	Replaces all occurrences of a string
<b>hist()</b>	Plot a histogram from a list of data
<b>install.packages()</b>	Downloads and prepares a packages for use
<b>lapply()</b>	Applies a function to a list
<b>library()</b>	Loads a packages for use
<b>lm()</b>	Fit linear model
<b>ls()</b>	List objects in current environment
<b>mean()</b>	Calculates the arithmetic mean of a vector combines
<b>median()</b>	Finds the statistical center point of a list of numbers
<b>nei()</b>	Returns all vertices which matrix are neighbors of specific vertex
<b>order()</b>	Returns a sorted list of index numbers
<b>paste()</b>	Concatenate vectors after converting to character
<b>plot()</b>	Generic function for plotting of R projects
<b>predict()</b>	Generic function for predictions from the results of various model fitting functions
<b>rainbow()</b>	Creates a vector of n contiguous colors



<b>range()</b>	Returns the minimum and maximum of value
<b>rbind()</b>	Combines vector, matrix or data frame by rows
<b>read.csv()</b>	Reads a file into data frame
<b>read.table()</b>	Reads a file into data frame in table format
<b>removeURL()</b>	Removes URL
<b>removeNumbers()</b>	Removes numbers
<b>removePunctuations()</b>	Removes punctuations
<b>removewords()</b>	Removes words from corpus
<b>rgb()</b>	Creates colors corresponding to the given intensities (between 0 and max) of the red, green and blue primaries
<b>rownames()</b>	retrieves or sets the row names of matrix
<b>rowSums()</b>	Computes the sums of matrix rows
<b>rpart()</b>	Is used to build a decision tree
<b>set.seed()</b>	Sets a fixed random seed
<b>str()</b>	Returns the structure of a data object
<b>stripWhitespace()</b>	Strips extra whitespace from a text document

<b>strwrap()</b>	Wrap character strings to format paragraphs
<b>sum()</b>	Adds up all elements of a vector
<b>summary()</b>	Produces an overview of the contents of a data structure
<b>t()</b>	Transposes a matrix or a data frame
<b>table()</b>	table uses the cross-classifying factors to build a contingency table of the counts at each combination of factor levels
<b>TermDocumentMatrix()</b>	Constructs or coerces to a term-document matrix or document-term matrix
<b>tm_map()</b>	Is an interface to apply transformations to corpus
<b>tolower()</b>	Converts string to its lowercase
<b>v()</b>	Access to the vertices of graphs
<b>var()</b>	Calculates variance of a list of numbers
<b>wordcloud()</b>	Plots a cloud of words shared across documents