DATA MINING FOR STUDYING THE IMPACT OF REFLECTION ON LEARNING

RAJIBUSSALIM, MInfoTech

A thesis submitted in fulfilment of the requirements
for the degree of Master of Science

Faculty of Science
University of Sydney
Australia

May 2014

Statement of Originality

This thesis contains no material that has been accepted for the award of any other degree in any university or other institution. To the best of my knowledge, this thesis contains no material previously published or written by another person except where due reference is made in the text of the thesis

Rajibussalim

21 May 2014

Abstract

On-line Web-based education learning systems generate a large amount of students' log data and profiles that could be useful for educators and students. Hence, data mining techniques that enable the extraction of hidden and potentially useful information in educational databases have been employed to explore educational data. A new promising area of research called educational data mining (EDM) has emerged.

Reflect is a Web-based learning system that supports learning by reflection. Reflection is a process in which individuals explore their experiences in order to gain new understanding and appreciation, and research suggests that reflection improves learning. The Reflect system has been used at the University of Sydney's School of Information Technology for several years as a source of learning and practice in addition to the classroom teaching. Using the data from a system that promotes reflection for learning (such as the Reflect system), this thesis focuses on the investigation of how reflection helps students in their learning. The main objective is to study students' learning behaviour associated with positive and negative outcomes (in exams) by utilising data mining techniques to search for previously unknown, potentially useful hidden information in the database.

The approach in this study was, first, to explore the data by means of statistical analyses. Then, popular data mining algorithms such as the K-means and J48 algorithms were utilised to cluster and classify students according to their learning behaviours in using Reflect. The Apriori algorithm was also employed to find associations among the data attributes that lead to success. We were able to group and classify students according to their activities in the Reflect system, and we identified some activities associated with student performance and learning outcomes (high, moderate or low exam marks). We concluded that the approach resulted in the identification of some learning behaviours that have important impacts on student performance.

Acknowledgements

I would like to first acknowledge and thank my wife Indah Sriwindhari and our daughters Aisha Fathiya Izzati and Nazia Hafizhatul Husna for their unconditional love, continuous support and encouragement. My special thank goes to my parent, who I can not thank enough for their unlimited support and pray with love for me and family.

I greatly appreaciate and thank to my supervisor Dr. Kalina Yacef for all invaluable guidance and support during my candidature. I would also like to thank my associate supervisor Professor Judy Kay who provides me with her insightful comments and advices.

Finally thanks to Dr. William Niu and Anindito Aditomo who gave valuable feedback in the completion of this thesis and to Greg Darke, Rafiuddin Palinrungi and Sardad Mohammad Farhad for helping me with the proof reading.

I gratefully acknowledge the Australian Government for providing financial support for my study.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Learning by reflection is considered to be important since learning effectiveness can increase when learners adapt to their learning experiences by reflecting on their learning processes and the state of their knowledge [14, 77, 78]. In this thesis, we define reflection as the process in which "individuals engage to explore their experiences in order to lead to a new understanding and appreciation" [14]. There are many different ways that people learn by reflection; one of these ways is self-assessment. Boud [13] defined self-assessment as "the involvement of students in identifying standards and/or criteria to apply to their work, and making judgment about the extent to which they have met these criteria and standards". Self-assessment is important because it can help students to develop the ability to identify their strengths and weaknesses and focus their study efforts on the particular area they believe needs improvement.

The present study focuses on how the data gathered from an online system that supports learner reflection, namely, the Reflect system at the University of Sydney, can be explored using educational data mining (EDM) methods. EDM is defined as "an emerging discipline, concerned with developing methods for exploring the unique types of data that come from educational settings, and using those methods to better understand students, and the settings which they learn in" [1]. The study aims to provide new insights into students' learning behaviours and information that can be useful for teachers and their students. This information is important because unlike teaching in

[1] http://www.educationaldatamining.org

classroom environments, where educators are able to give and receive direct feedback from students regarding their performance and learning experience, it is difficult for educators to gain this type of information from an online teaching environment. It is even more difficult for educators to continuously evaluate the students' progress and performance. Hence, the application of EDM methods to educational data as demonstrated in this study also represents an attempt to address these issues. The information and knowledge resulting from this study can be useful for teachers to better understand their students' learning behaviours and to inform students if their current behaviour is associated with negative or positive outcomes. The information can also be beneficial if presented to students as it enables them to consider how effective their learning habits are against their current performance.

1.2 Motivation and Study Objectives

The objective of EDM research is to provide a deeper understanding of the key factors that impact on learning. One way to achieve this is by investigating students' learning behaviours; in particular, how students perform self-assessment and reflect on their learning behaviours in ways that affect the learning outcomes. There is considerable research on students' learning behaviours but there has been little research on the use of EDM techniques to investigate the relationships between reflection and learning outcomes. This concern has motivated the undertaking of the present research. In particular, the present research focuses on the identification of students' behaviours related to the use of the Reflect system for reflective learning that resulted in positive learning outcomes for the students. At the same time, the research investigates how students' performances can be evaluated based on these criteria so that educators can identify students at risk of poor learning outcomes.

This thesis focuses on the use of EDM methods to extract information about how reflection helps students in learning by studying data from the Reflect system. In particular, the main objective is to evaluate the effectiveness of EDM methods for: (1) gaining knowledge about students' learning behaviours; (2) identifying which behavioural patterns lead to positive or negative outcomes; and (3) extracting knowledge about the impacts of reflection on learning.

In order to achieve these objectives, popular data mining techniques are utilised including clustering, classification, and association rules mining. The importance of this study is indicated by the conclusion made in a large number of previous studies that reflection improves learning.

1.3 Research Question

Based on the study objectives that highlight the importance of reflection for learning, the primary research question is defined as follows:

Can mining educational data help to better understand the impact of reflection on learning?

Addressing this research question requires theoretical and analytical studies of the applications of EDM methods on the Reflect data. The analysis requires the examination of the behaviour of students when learning, in particular when learning by using a system that supports learner reflection (i.e., the Reflect system) and its impact on their performance.

The primary research question forms the framework for this study. However, this main research question can be restructured to be more specific and focused. Therefore, the research question is translated into more specific subsidiary research question as follows:

Do students learn better by using the Reflect system?

In order to address this subsidiary research question, a number of hypotheses are developed as discussed in Chapter 4. The main reason for translating the subsidiary research question into the hypotheses is because a hypothesis can be directly tested; hence, its results can be measured and proved or unproved. The main hypotheses (H1 - H3) are related directly to the use of the Reflect system as a means of reflection and are tested by using the dataset from the Reflect system, while another two hypotheses (H4 and H5) are developed and tested by using the student assessment data from the relevant course coordinator. In order to test these hypotheses, a series of experiments was conducted. The results of the experiments provide answers to the primary research question and its subsidiary research question.

1.4 Research Contributions

Within the scope of the research goals, the contributions of this thesis are as follows:

- The study provides insights and valuable information about how EDM methods can be used to extract information from the data gathered from a system that supports learner reflection.

- The study includes a detailed analysis of the results that lead to the identification of students' learning behaviours related to the use of the Reflect system (three discrete groups of users are identified).

- The study identifies the students' learning behaviours in the Reflect data that lead to positive and negative outcomes.

- The study provides analysis about the correlation between the diverse self-assessments tasks (reflective learning) performed in the Reflect and the students' performance.

- The results of the study provide some pedagogical knowledge that can be useful for teachers regarding the learning behaviours associated with positive and negative outcomes; this knowledge can be used by teachers and students to improve study performance.

1.5 Thesis Outline

This thesis is organised in six chapters.

Chapter 1 introduced the key concepts of learning by reflection, the motivation behind this study and the research objectives. The research question was defined and an overview of the research contributions was presented.

Chapter 2 presents the review of the literature in the area of educational data mining. It discusses the issues related to the use of EDM methods to explore educational data.

Chapter 3 discusses the issues related to learning by reflection. It presents an overview of the Reflect system including how it supports learner reflection and its procedures of self-assessment.

Chapter 4 describes the research approach and methodology, including the research question and hypotheses, the subject and the data used in the study, the process of data mining and the application of EDM methods to the data.

Chapter 5 presents the results of the empirical studies set up in Chapter 4, followed by a discussion about the results. Useful information about students' learning behaviours in using the Reflect system is presented.

Chapter 6 provides the conclusion of the thesis. The limitations of this study are discussed, and potential areas for future work are recommended.

Chapter 2

Educational Data Mining Research

This chapter provides an overview of educational data mining through a review of prominent work in the EDM research community that is related to and provides a guideline for the work in this thesis. This chapter highlights the positive aspects of the previously published work and identifies potential problems that can be addressed in the present research and in future work. This chapter also describes the purpose of EDM research, the types of data used in EDM research (including typical educational systems that generate this data) and the extraction of pedagogical aspects (EDM for studying learning behaviour). The chapter is concluded by a review of the most popular EDM methods and techniques currently in use.

2.1 Introduction

The emergence of computing technology provides educational institutions with new alternatives for the delivery of learning materials to students, reducing the reliance on face-to-face teaching in classrooms. There is a current trend towards the management of teaching and learning activities through the use of online web-based learning systems that are delivered through the Internet [75].

One advantage of online web-based learning systems is their ability to record data about most of the activities done by the students in the system; hence providing detailed learning profiles [69] that can be mined. Many institutions that utilise these systems are able to automatically collect large volumes of students' interaction data. Within the last two decades, researchers have realised that this data can be useful to both educators and students. Thus, *data mining techniques* that enable the extraction of hidden and potentially useful information in the data have been used on educational data. Popular data mining techniques such as *clustering*, *classification*, *association rules mining* and *sequential pattern mining* have been applied to the educational data with promising results.

The term of *data mining* is defined as "a step in the overall process of knowledge discovery in databases that consists of pre-processing, data mining, and post-processing" [89]. Data mining also encompasses the *Knowledge Discovery in Database* (*KDD*) technique which is defined as "extracting or mining knowledge from large amounts of data" [32]. The use of data mining techniques to analyse educational data is a growing area of research as it supports the analysis of learners' behaviours using data acquired from online web-based learning environments [16, 25, 31, 44, 45, 62, 73, 79]. This emerging research area is known as *Educational Data Mining* (*EDM*) [33, 52, 69, 75]. As a new research area, the main focus of EDM is exploring the unique types of data that come from educational systems and tools. This is done by developing methods to support students and the educational settings they use[1].

The abundance of data but lack of powerful data analysis tools has been described as "*a data rich but information poor situation*". As a result, important decisions are often made based not on the rich information retrieved from a database but on intuition of a decision maker. This is because the decision maker does not have or is not capable of extracting rich information or knowledge from the data[33]. Currently, EDM research integrates the interdisciplinary research fields of Statistics and Visualisation, Psychological Education, Knowledge Discovery and Database, Machine Learning, Information Science, and Artificial Intelligent [38, 69, 75]. The main reason behind the rapid development of EDM research in recent years is due to the availability of enormous amounts of educational data, mostly generated by web-based educational systems, and the urgency for converting such data into useful information and knowledge for decision making [33].

EDM researchers consider this phenomenon as an opportunity to explore data mining models that can predict the future behaviours of learners. This information

[1] http://www.educationaldatamining.org

can be used to enhance the decision-making process.

2.2 Purpose of EDM Research

Researchers have recently begun to explore the possibilities of applying data mining techniques to educational data. This research has mostly been intended to help educators and tutors to better understand their students' learning behaviour by allowing them to assess the students' performance and track their learning progress [26].

Another goal of EDM research has been to provide insight into the parts of a course structure that need to be revised in order to improve learning. Research of this type is primarily intended to help educators [16, 51, 52, 54, 65, 75]. Other EDM research has been oriented towards the provision of benefits for both teachers and their students [38, 36, 70, 69].

The following sections present a review of the research in EDM that serves all of the above purposes.

2.3 Overview of EDM Research

Research in EDM can be categorised according to the types of educational data used for mining and the data mining techniques used to mine the information within these data. This section discusses the types of data that can be used in EDM research and followed by an overview of EDM methods used to mine information from various web-based educational systems.

2.3.1 Types of Data Usedin EDM Research

The data gathered from online web-based educational learning systems and tools comes in different formats that are specific to that particular tool. It is certainly possible that the data generated from an educational system is also stored into a database. But the types of data it stored depend on the setting on the database itself because modern databases are capable of storing multiple formats including .dat, .mdl, .text, or log file itself. For example, an educational system may generate data in one of the following forms: a text format, web server log file, or LMS log files [11]. Each of these data formats used in EDM research is briefly discussed as follow.

Text Data

This type of data is characterised by its rich and high-level information.This data is difficult to use for data mining as it requires a considerable amount of time to hand code [11]. Several researchers [21, 43, 49] have employed data mining on this type of data in both LMS and Computer Supported Collaborative Learning (CSCL).

Web Server Log Files

Web server log files contain vast collections of data that are produced when users' access specific web pages [33]. The characteristic of this low-level data that is collected in the server logs is a high level of noise. This noise makes the date difficult to organise and often require intensive coding before the data can be used for mining [11].Various data mining techniques have been employed on this type of data in order to extract useful information from it [39, 41, 57, 90, 91].

LMS Log Files

Learning Management System (LMS) log files require the least amount of hand coding as they keep track of users' login and sessions and also store high level data such as students' grades and messages they have posted in the system [48]. Data from LMS log files has been used by many researchers for data mining [28, 63, 75, 74, 87].

Data from User Models

A *user model* (UM) or learner model represents the characteristics of a user in the system so the system is able to distinguish each user in the system. To build a UM, a system requires data about the user. This data can be obtained when the user logs into the system or when directly requested of the user. There are different types of user data commonly used to build a UM, including the users' characteristics (such as gender, age, marital status and location), users' preferences and interests, the users' knowledge and skills, and the users' behavioural patterns [40].

The present research employs data mining techniques on the data collected by the Reflect system. The Reflect system uses a user model to capture information about the user. All of the infomation about a user is recorded into his or her user model, thus enabling the system to monitor the state of the knowledge of the users and their learning progress [36].

This leads to the proposal of the following questions "Can the Reflect data be mined?" and, "if so, How difficult is it to do?" Previous work has reported that mining the data generated by an online educational system such as LMS or ITS (Intelligent Tutoring System) is challenging [63] and requiring different data mining tools and techniques depending on the types and granularity of the data. This is made more difficult as the Reflect system was not designed to accomodate the data mining applications.

2.3.2 Applications of EDM Methods

Nowadays, enormous amounts of educational data are generated from online web-based educational and e-learning systems. These systems are either used as a supplement to normal classroom teaching methods or as a sole teaching and learning medium for courses such as in distance learning programs. Most EDM researchers have used this data by applying EDM methods to the data generated by an online web-based educational system.

A number of current EDM methods were reviewed by Baker and Yacef [6] in the first issue of Journal of Educational Data Mining (JEDM) 2009[2]. They identified the two most predominant categories of EDM methods as follows: a category of methods based on viewpoint of Romero and Ventura [69] which consist of:

- Statistics and visualisation
- Web mining, that can be further categorised into:
  - Clustering, classification, and outlier detection
  - Association rule mining and sequential pattern mining and
  - Text mining

According to Baker and Yacef [6] the web mining methods listed by Romero and Ventura are quite prominent in contemporary EDM research.

The second predominant category of EDM methods was proposed by Baker [3] as follows:

[2] http://www.educationaldatamining.org/JEDM

- Prediction
  - Classification
  - Regression
  - Density estimation
- Clustering
- Relationship mining
  - Association rule mining
  - Correlation mining
  - Sequential pattern mining
  - Causal data mining
- Distillation of data for human judgement
- Discovery with models

Prediction, clustering and relationship mining are quite popular among EDM researchers while the fourth and the fifth categories according to Baker and Yacef [6] are justified by having a prominent place in published EDM research. Examples of research into "distillation of data for human judgement" are [37] and [85]. Examples of "discovery with models" are [9], [19] and [34].

The EDM methods listed in the first two categories discussed above have been the most prominent EDM methods used by researchers within the last decade. For example, these EDM methods have been widely applied to educational data gathered from learning management systems, intelligent tutoring system, and online collaboration tools. To understand about how these methods have been utilised in EDM research, the key application of the most popular EDM methods is discussed in the following sections.

EDM Methods Applied to LMS Data

A Learning Management System (LMS) is a web-based learning system that offers great flexibility in terms of time and space for information exchange in learning and teaching. There are many types of LMS currently used at educational institutions. One of the most popular and widely accepted LMS is Moodle (Modular Object Oriented Developmental Learning Environment). Moodle has been used by many organisations to produce web-based courses and web sites [58]. Moodle also helps to facilitate information sharing and communication between members of a course and its teachers. The main benefit of the Moodle system is that it allows teachers to perform many tasks such as preparing lecture materials and tutorial questions, preparing for assignments, quizzes and exams, and also can be used to set up online collaboration media such as chat rooms, discussion forums and online feedback [58].

Moodle can generate a vast amount of data related to the various activities done by users such as accessing lecture materials or assignments, taking a test, reading questions, browsing the web, writing solutions and many other tasks including communicating with teachers or other users through a chat room. This kind of information is considered very valuable in the study of how students behave when interacting with the system and to discover the possible relationships between the various tasks carried out in Moodle and the students' final mark. The right data mining algorithms may be used to discover these relationships as suggested by Mostow [60].

A number of researchers in the EDM field have utilised the data generated from the Moodle system [7, 68, 75]. Romero et al in [68] and [75] have applied various data mining techniques to mine the data generated by the Moodle system. In [75], data was collected from 438 students across seven courses that used Moodle as an alternative learning source to face-to-face teaching at the University of Cordoba, Spain. These students were chosen because they were among the most active students using Moodle; for example, they used the system to do assignments and quizzes and to communicate in chat rooms and forums. The activity data and the students' exam marks were summarised in a table. This table also summarised the total time each student had spent on each task. To increase interpretation and comprehensibility, a discretisation table was also created. The discretisation table was used to represent the numerical data into separate categories that were easier to understand. For example, the numerical marks and attributes were classified into four intervals, namely, FAIL if the value is < 5, PASS if the value is >5 and <7, GOOD if the value is >7 and <9, and EXCELLENT if the value is >9. All the other attributes were transformed to three equal-width intervals labelled as LOW, MEDIUM and HIGH. The final data pre-processing step was to convert the data into a suitable format to be used with data mining tools. Since this research used Weka, the data was transformed into the attribute relation file format (ARFF). This is an ASCII text file suitable to be used in Weka [89].

Next, the K-means algorithm was used to cluster students from a particular course into different groups depending on their activities in Moodle and their final marks. The course that was chosen to be data mined was 218 (Technical Office). The K-means algorithm was performed with the value of k (number of clusters) equal to 3. As a result, students were grouped into three clusters in which each group had a different characteristic as summarised in Figure 2.1. The clusters of students were named "very active students" (Cluster 1), "active students" (cluster 2) and "non-active students" (cluster 0).

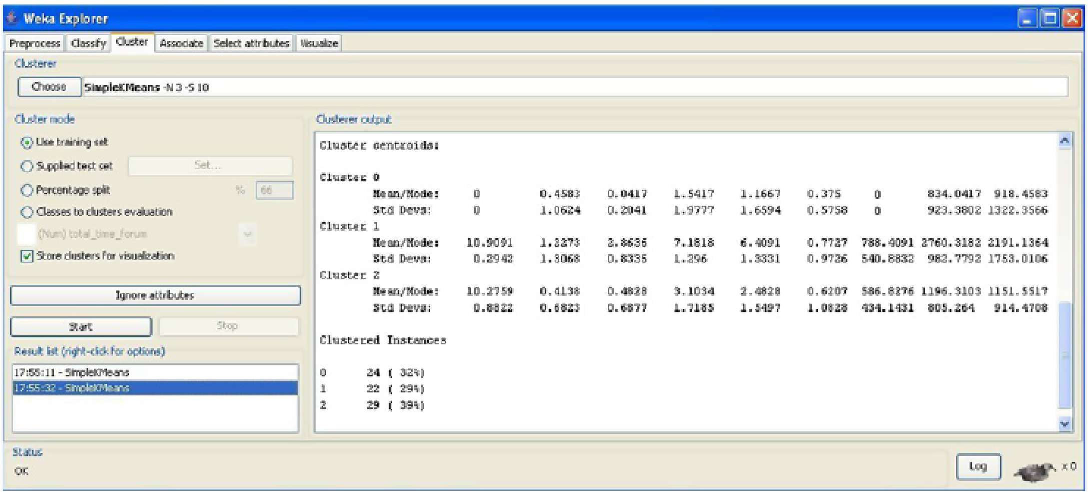


Figure 2.1: K-means algorithm executed in Weka (from [75])

The very active students were characterised by very active involvement with the Moodle system such as a high number of quizzes done and passed (7.1 and 6.4), a low number of failed quizzes (0.7), and a high total number of time spent on assignments, quizzes and forums. Non-active students behaved in the opposite way to the very active students, such as no assignments done (0), very low number of messages read (0.045), very few quizzes done and passed, a high number of failed quizzes (1.54, 1.16 and 0.37 respectively) and low total time spent on assignments, quizzes and forums. The active students' behaviours were in between those two groups.

The researchers argued that this information can be used by teachers or instructors of the course to form study groups for collaborative activities with the students from different behavioural groups or to group new students into the clusters depending on their characteristics [69].

The literature has also reported that classification and association rules mining have been applied to analyse the student data generated by the Moodle system. The classification algorithm (the C4.5 algorithm) [67] was used for classifying students into several groups with equal final marks depending on their Moodle activities while an Apriori algorithm was implemented to search for association rules over the discretised summarisation table for the course 110 (Projects). As a result, a decision tree was generated (Figure 2.2) with a summary of the number of nodes and the number of leaves on the tree, and the number and the percentage of correctly and incorrectly classified instances. The decision tree revealed the rules that classified students into three categories, namely, FAIL, EXCELLENT and MEDIUM (FAIL, PASS or GOOD). Students were labelled FAIL if they had a low number of passed quizzes, while students who had a high number of passed quizzes were directly labelled EXCELLENT. The rest of the students who earned a medium number of passed quizzes were classified as FAIL, PASS or GOOD depending on other values such as total time spent on assignments, numbers of quizzes, number of quizzes failed, number of assignments, and number of courses .

Researchers concluded that by using the information discovered by these rules, educators can make decisions about Moodle course activities and use the rules for classifying new students.

Meanwhile, the Apriori algorithm that was first introduced by Agrawal et al. [1, 82] was used to generate association rules mining. The algorithm that is build-in and freely available in Weka was executed with a minimum support of 0.3 and a minimum confidence of 0.9 as parameters. As a result, a number of interesting rules were discovered as shown in Figure 2.3. However, several similar rules were also discovered (the rules with the same element in antecedent and consequent but interchanged). Some interesting facts are that some rules revealed relevant information for educational purpose but others formed unexpected relationships. The authors argued that this information can be very useful for educators in making correct decisions about the current students' activities and for monitoring students with learning problems.

Although the Reflect system is not a learning management system and does not provide the functionalities for information sharing and communication that an LMS such as Moodle does, it does collect data about the users' interactions that is partially

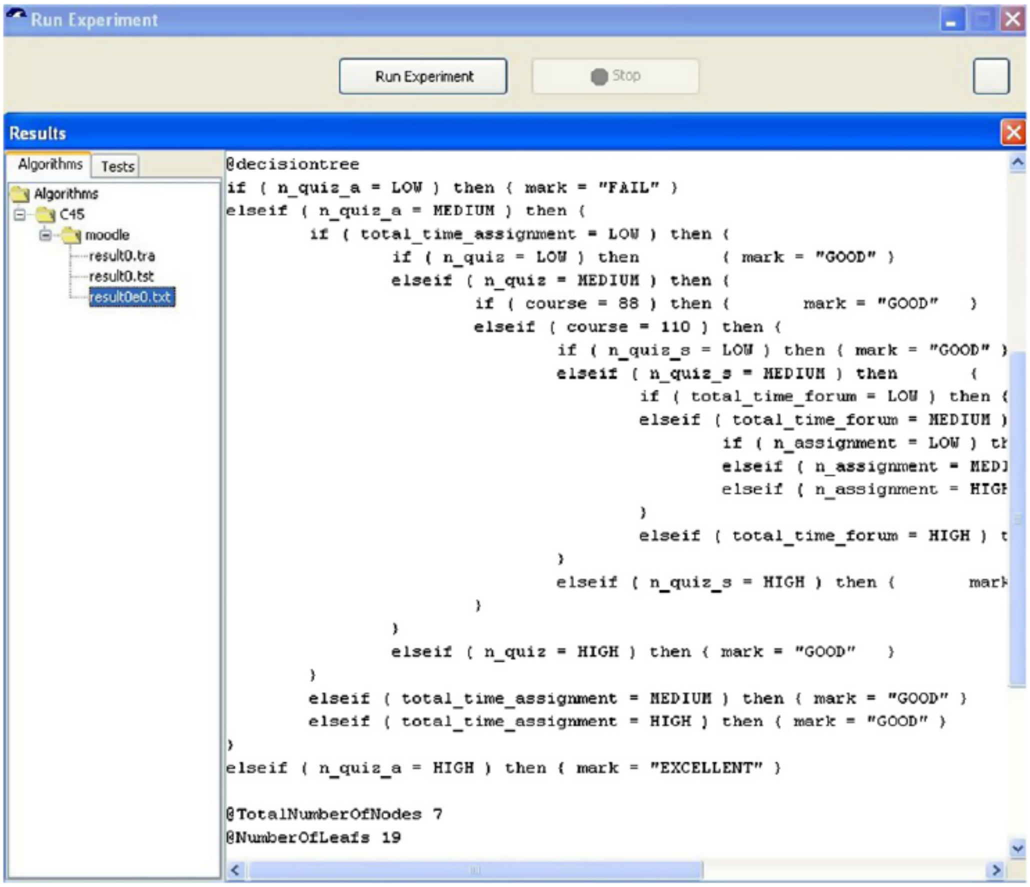


Figure 2.2: C.45 algorithm executed in Keel (from [75])

similar to the data collected by the Moodle system. For example, it collects data about students' profiles and students' task submissions. In addition, some degree of similarity is found in the approach adopted in the present study and the LMS data pre -processing techniques.

EDM Methods Applied to ITS Data

Nowadays, Intelligent Tutoring Systems (ITSs) are widely employed by many educational institutions such as universities, colleges and schools. This popularity is due to the facts that an ITS can record almost every single click of a user's interaction with the system. The data commonly recorded by an ITS is in the form of log files. This is because the log file is easy to record and offers flexibilities in terms of the type of information it can capture . A number of EDM researchers have utilised the data generated from an ITS to conduct their experiments. For example, Feng et al [26] used

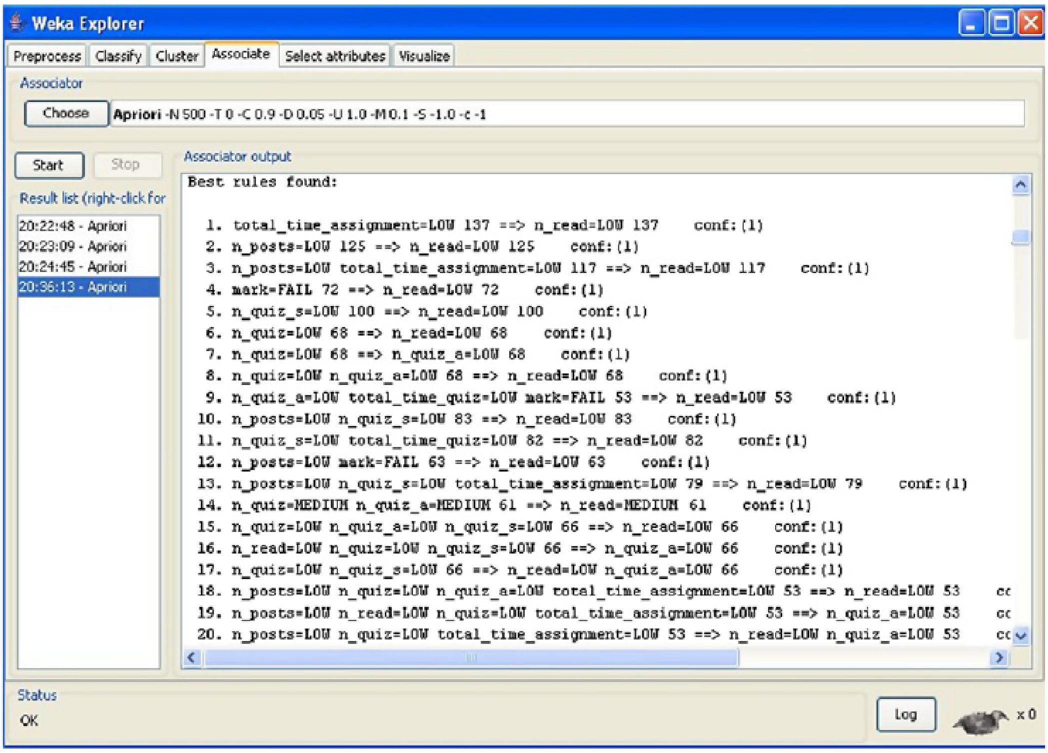


Figure 2.3: Apriori algorithm executed in Weka (from [75])

data generated from an ITS system called ASSISTments to evaluate and predict students' Math proficiency compared to a state standardised test. Matthews and Mitrovic [47] also used data generated by an SQL tutor to determine the relationships between the number of constraints seen in an ITS and the number of constraints learnt. Other ITSs that generated educational data and have been used in EDM include: Geometry cognitive tutor ShihKS08 [81], Project LISTEN's reading tutor [59]and [61] and Logic ITA [51] and [52].

However, not all the data gathered from an ITS is mineable. Some data is unable to be mined because the ITS was not designed for data mining purposes in the first place. ITS log files can be successive in the number of instants (users) and sessions. They can also be very rich in the level of detail . The attributes that make an ITS mineable are *multiple grain size*, *reifying tasks*, and *timeable* [59]. Multiple grain size data supports multiple types of analyses. For example, the duration and frequency of student sessions are among the attributes required to measure the usage of an ITS while the data at individual read word is useful for much finer-grained analyses. Reifying the task means that the data produced by an ITS must be *machine-understandable* that is the data is able to be processed and computed by machines. Timeable means that the data is observable over an interval of time. This is because not all activities are observable in terms of time; for example, a student's reading comprehension process is mental and therefore it is difficult to be observed accurately in terms of the time spent.

The Reflect system offers tutoring functionalities that are similar to an ITS. It teaches students programming and provides a number of solutions which students can use to compare their own work. From the Reflect system, a student might learn the requirements for being a good programmer, such as what are the better indentation styles, how to write better comments and other similar concepts. However, the more important aspect of ITS research is that it provides insights into how educational data gathered from an ITS is mined.

EDM Methods Applied to Online Web-Based Collaboration Data

The increased use of online collaboration tools and Computer Supported Collaborative Learning (CSCL) within educational institutions has resulted in the availability of a large amount of educational data. Researchers see this as an opportunity to mine the data in search for useful information and knowledge. For example, Perera et al. [65] applied various data mining algorithms such as the K-means, EM clustering and sequential pattern mining algorithm to the data collected by Trac, an online collaboration tool used for a senior software development project course, to distinguish strong students from the weak ones and to search for patterns of leaderships and collaborative work among the team members.

The K-means clustering with the number of clusters set at 3 (k=3) was applied to the students' data both as a group and as individuals. From clustering the groups of students, it was revealed that many attributes had a high correlation. As a result, the manual composite attributes were created. One preferred attribute was chosen over another; for example, how Trac was used and when it was used were favoured over the total activity. The end result, after the new attribute selection, is presented in Table 2.1.

The results suggest that Group 1 behaved very differently to Groups 2, 3, 4 and 6, consisting of students who frequently used Trac. It indicated that high level activities in Trac such as ticketing and SVN activities, and high events per wiki pages set these

Table 2.1: Group clustering results (from [65])

Clustered groups	Distinguishing characteristics
Groups 2, 3, 4 & 6	Moderate events per ticket, Infrequent Trac activity (tickets and SVN), Moderate % of ticket update events, Moderate number of lines added/deleted per Wiki edit
Groups 5 & 7	Moderately frequent Trac activity (tickets and SVN), High edits per Wiki page, Low number of lines added/deleted per Wiki edit, Low number of events per ticket, Low % of ticket update events
Group 1	Very frequent Trac activity (tickets and SVN), High events per Wiki page and per ticket, High Wiki page usage span, High % of ticket update events, High % of ticket accepting events

students apart from the students in the other groups.

In the next analyses, the researchers performed clustering analyses on the students individually. By clustering students individually, more interesting facts were discovered from the data as shown in Table 2.2 and 2.3.

As shown in Table 2.2, the researchers were able to label individual students according to their activities in the Trac system. For example, a very active student was labelled as a 'Manager' while a less active student was labelled as 'Majority'. In Table 2.3, the researchers demonstrated that some groups could be differentiated from others by the absence of a manager, explaining the leadership problems these group had encountered.

The research also tried to explore more information from the Trac data regarding the timing of an action (event) in relation to other actions (events). The researchers believed that certain sequences of events characterised the better groups from the weaker ones. This information was obtained through the application of sequence pattern mining algorithms to the data from Trac.

According to the authors, some new findings were discovered through the applications of sequential pattern mining. For example, it was revealed that among the top groups, tickets were used more frequently than the Wiki; while in the weakest group it was the opposite. This might have suggested that the use of the ticketing system referred to the actual work being done as it was more task-oriented than the Wiki. The authors argued that the result was consistent when compared to the behaviour of the strong group leaders who used more tickets than the leaders of the weaker groups.

Table 2.2: Group clustering with modified attribute (from [65])

Clustered size	Distinguishing characteristics	Cluster label
8 students	High ticketing activity, Involved in many tickets, High Wiki activity, Involved in many Wiki pages, Moderate SVN activity	‘Managers’
9 students	Moderately high ticketing activity, Ticketing occurring on many different days, Moderate Wiki activity, Very high SVN activity	‘Trac-Oriented Developers’
11 students	Low ticketing activity, Low Wiki activity, Low SVN activity	‘Loafers’
15 students	Moderately low ticketing activity, Moderately low Wiki activity, Many Wiki events on days which Wiki events occurred, Many SVN events on days which SVN events occurred	‘Majority’

The authors also suggested that the top group leaders were less involved in technical work and delegated the tasks properly to other group members, while the leaders of the weaker groups tended to either be involved in a lot of technical work or they used the Wiki more than the tickets.

This work by Perera et al. is directly related to the present thesis in two ways. Firstly, as in this thesis, their subject is the behaviour of the learners (students) in an online web-based learning environment. Secondly, the EDM methods and approach used in the present study have some degree of similarity with the work by Perera et al.; for example, the K-means and decision tree algorithm are also used here to separate a strong group of students from the weak ones.

Table 2.3: Students distribution from each cluster (from [65])

Clustered size	Managers	Trac-Oriented Developers	Loafers	Majority
Group 1	*1	3	1	1
Group 2	*1	0	1	3
Group 3	0	1	2	**3
Group 4	*1	3	2	0
Group 5	3	*1	0	3
Group 6	*1	1	3	1
Group 7	*1	0	2	4

2.4 Students' Learning Behaviour

The identification of students learning behaviours, in particular those behaviours that distinguish or characterise students, is important and can reveal important information about the students. This information can be used for monitoring the students' learning progress [79, 34], predicting the learning outcomes and students performances [42, 26] and identifying successful learners [50, 47] and students at risk [8, 20]. Therefore, many EDM studies have focused on this area. One important aspect of students' learning behaviours that recently began to attract more attention is self-regulated learning (SRL). Some recent studies have focused on this area of students' learning behaviours [12, 76, 42, 50]. A number of studies have related SLR to *self-assessment* [50, 76]. Self-assessment is an importance learning behaviour that can be used to improve students' understanding of concepts and results in the increased chance to achieve a better learning outcome. In the current research, the concept of self-assessment is seen as how students interact and participate in an LMS or ITS. The amount of interaction performed by the students in an educational system (for example viewing concepts, reading and replying to the forum, posting messages, or interacting with other students) is calculated in order to determine the level of self-assessment the students had performed.

The present study views the concept of self-assessment differently from the existing research. In this study, self-assessment is viewed as a direct action by students to self-assess their understanding of a learning task directly in the educational system. The students can rate their understanding of a programming concept based on certain criteria set up by the course coordinator such as "Very Well", "Well" "Ok" or "Not at all". Hence, the students' self-assessment participation that is used to measure how much self-assessment has been performed by the students (the total self-assessment) is calculated from the number of times the students have rated themselves in the learning educational system. This self-assessment procedure is possible because the present study uses the data from the Reflect educational system that supports learner reflection. In this regard, the concept of self-assessment in the present study is different to the concept of self-assessment of the extant research in EDM. A more detailed discussion about the self-assessment procedures in the Reflect system is presented in Chapter 3.

2.4.1 Factors that May Interfere with Learning Progress

A number of studies have indicated that several factors may interfere with learning progress. These factors include but are not limited to: (1) previous background knowledge of the topic, (2) lack of motivation and engagement in the learning processes, and (3) possible misuse of the system.

Background Knowledge

A number of studies reported that there is a significant advantage for students who have prior knowledge in a programming language to be successful in their first year programming courses [29] [15]. One of those studies [15] indicated a positive relationship between prior knowledge in Mathematics or Science to the success in computer programming studies. The researchers believed that there is a degree of similarity between the concepts for learning Mathematics and learning a programming language. As a result, a student who had learnt those concepts previously might find them easier to comprehend and make a better progress in learning other concepts. In contrast, a student who had not learnt the concepts beforehand will require more time to grasp similar concepts.

Lack of Motivation and Engagement

Motivation is the most important component in learning (Bandura, 1986 as in [18]). There is a large body of literature devoted to this topic. In general, motivation is related to the questions about why students engaged, performed and completed certain tasks.

Other researchers [56] have reviewed the six motivational constructs in a traditional face to face classroom. These constructs have not been investigated in an online environment. These six motivational constructs are (a) self-efficacy, (b) locus of control, (c) attributions, (d) goal orientation, (e) intrinsic versus extrinsic motivation, and (f) self-regulation. Researchers have concluded that motivation is considered the most important factor in learning because: it increases individual's energy and activity levels [46], it can direct an individual toward certain goals and achievements [23], it promotes the initiation of certain activities and persistence in those activities [83], and it affects the learning strategies and cognitive processes that individuals employ [24].

Misuse of the System

The progress of learning may also be affected when students misuse the system. The phrase of "*misuse the system*" or "*gaming the system*" is associated with the behaviour aimed at obtaining correct answers by systematically misusing the software's help and feedback without having to know why the answer was correct [5].

Baker et al. in [5] and [4] indicated that students who frequently misused the system learnt only two thirds as much as other students who did not engage in such behaviour. In their research, Baker at al. also discovered that students who gamed the system had low knowledge in the pre-test and had low overall academic achievement.

2.4.2 Methods to Study Students Learning Behaviour

Common methods to study students' learning behaviour include case studies, surveys and interviews.

Case Study

A case study is an examination of a specific behaviour such as an events, activities, institutions or group of person [55]. Research using case studies allows the researcher to study and evaluate the behaviour and phenomena in its natural environment. A case study can employ multiple sources of evidence gained from various sources including a person, groups or institutions.

Case studies have been used in EDM research to study students' learning behaviour. For example, Pechenizkiy [63] used data mining techniques to study effectiveness of immediate tailored and elaborated feedback (EF) for students taking an online exam. In that study, the researcher used a small number of participants (73 students). The results suggested that it was difficult for researchers to obtain meaningful results with a set of traditional data mining techniques including clustering, classification and association rules mining even with a small amount of data.

Survey and Interview

The survey approach is a type of research that collects data which aims to capture snapshots of practices and situations at a particular point in time [27]. The advantage of this type of research is its ability to utilise and observe a large number of variables in the target population [27]. Some researchers have utilised this technique to study learning behaviour. For example, Sheard et al conducted a survey to study the impact of web-based learning environments on learning experiences and learning behaviour [80].

2.5 Extracting Pedagogical Information from EDM Research

One purpose of EDM research is to discover pedagogical aspects of educational data to help educators in decision making [57, 69, 86]. Previous work by Merceron and Yacef [51, 52] utilised various tools and data mining techniques including K-means clustering, hierarchical clustering and association rules mining with a modified version of the Apriori algorithm. The objective of their work was to search for interesting pedagogical information in data generated by Logic-ITA, an ITS used to help students practise logic formal proofs and to inform the teacher of the class progress. This tool was used to collect the interaction data of around 860 students' for a period of four years.

First, association rules mining was used to identify mistakes that often occurred together when students were trying to solve problems. The researchers built a modified version of the Apriori algorithms available in TADA-Ed, a platform that allows visualisation and mining of the educational data [10, 53], in order to find the relationships in the data. From the students' interactions data in 2004, the researchers discovered that the first rules indicated if students made the mistake of "*Rule can be* *applied but deduction incorrec*t" while solving an exercise. The authors claimed that this finding was quite similar across the other years (2001 to 2003) and suggested that the information can be used by teachers to review the course materials and ensure certain subtleties in the concepts are explained to students.

Furthermore, the researchers applied two distinct clustering algorithms namely K-means and Hierarchical clustering to identify students with difficulties from the interaction data in Logic-ITA. The K-means algorithm was implemented in TADA-Ed while the Hierarchical clustering for data mining was implemented in the SPSS Clementine system. The clustering analyses generated three clusters of students with difficulties: cluster 0 consisted of students who made many mistakes per exercise and did not finish the exercise, cluster 1 contained students who made only a few mistakes and the students who made medium number of mistakes were grouped into cluster 4.

Finally, the researchers used a decision tree algorithm to predict students' exam marks using the dataset from the previous year as a training dataset. The attributes that were chosen were the number of mistakes, number of exercises, difficulty of the exercise, number of concepts, and final marks. A decision tree was built from these attributes. The tree was then used to predict the students' exam marks according to its attributes. The model was then used in the following year to predict the grades obtained by new students in that year. The authors claimed that they had successfully extracted useful pedagogical information that can be used for decision-making by educators. For example, a student needs to do at least two exercises to encourage them to do more and be able to complete the exercises. By successfully completing the two exercise barrier, they were assumed to have grasped important concepts of the topic and were likely to be able to answer the questions related to those concepts in the final exam. The pedagogical knowledge extracted from that study can be used by teachers to help students improve their performance. The approach taken in that study can be adapted or used as a guideline for the present study.

2.6 Data Mining Techniques Commonly Used in EDM

The most commonly used data mining techniques in EDM are clustering, classification and association rule mining. By employing these data mining techniques in the area of educational data, it is expected that some useful information and knowledge can be extracted from the data. Data mining techniques used for mining educational data can discover information that can be used in formative evaluation, that is, to assist educators to establish a pedagogical basis for making a decision when designing or modifying a course or teaching methodology [69]. Data mining techniques may also be used to predict or trace a student's performance from the data gathered from either a traditional classroom or a web-based educational system. This section presents a general overview of the data mining techniques commonly used in EDM research. The data mining algorithms used in EDM experiments are briefly introduced, namely, the K-means algorithm for clustering, the J48 algorithm for classification, and the Apriori algorithm for association rules mining experiments. A more detailed discussion of the algorithms and how they are implemented in the experiments is presented in Chapter 4 regarding the research approach.

2.6.1 Clustering

Clustering, also known as an unsupervised classification, is a process of grouping objects into classes of similar objects so that the objects within a cluster have high similarity but are very dissimilar to the objects in other clusters [22]. In EDM, clustering has been used in a number of ways, including: grouping students involved in an online collaboration tool called Trac, in order to identify leadership patterns among students working as a team and to classify students' behaviour associated with positive and negative outcomes [65]; grouping students according to the mistakes they made [53]; discovering and grouping students with difficulties [52]; clustering students according to their response correctness [63]; finding groups of students who have similar learning characteristics; and encouraging group-based collaborative learning to provide incremental learner diagnosis [84].

The present study uses a standard implementation of the K-means algorithm available in the Weka data mining package called SimpleKMeans [30]. The algorithm is chosen for two reasons: it is a well-known and reliable clustering algorithm that is easy to use, and it is freely available in the Weka data mining package. The K-means clustering algorithm works based on a partitioning algorithm that organises the objects of a dataset into a number of specific groups or clusters. It uses a centroid-based technique to create clusters of the objects. The centroid is a central point that can be defined in various ways such as by the mean or by the medoid. The difference between two clusters is measured by the distance (Euclidean distance) of its centre points.

The objective in using clustering in the present study is to be able to identify the group of students who used the Reflect system the most. In this regard, the final objective is to investigate whether there are any relationships between the number of activities performed in Reflect and the students' final exam marks. The experiments use the numeric dataset related to student activities in Reflect. The dataset is pre-processed and a summary table consisting of four data attributes related to students' activities in Reflect is created. The chosen attributes contain information about how students used the Reflect system for enhancing their learning experiences; for example, how many times students submitted the tasks or examined the learning objectives set by the tutor in the system, how many times students performed self-assessments, and how many times students submitted correct evidence (called the results from local test evidence). Section 4.4.2 presents the research methodology and discusses the data used for the experiments and the clustering experiments in more detail. The results of the clustering experiments are presented in Section 5.1.2.

2.6.2 Classification

Along with clustering, classification is one of the most frequently studied problems in the area of data mining and machine learning [17, 74]. It involves predicting the value of a (categorical) attribute (the class) by using a model or a classifier. A model can be built by using a training dataset and verified by using a test dataset. Within the model, there is a class attribute that will be used to label a newly encountered, still unlabelled pattern (class). A classification technique works in a two-step process, namely, the learning step and the classification itself. In the learning step, a classification model is constructed by using a training dataset, while in the classification step, the model is used to predict the class label for a given dataset [32].

Classification techniques have been widely used in EDM, including: classifying students into different groups with equal final marks depending on their activities carried out in a learning system [75]. In this case the classification task is to classify or group students into three classes (of exam mark) that are high, moderate, and low; discovering that students who read elaborated feedback for a particular question increased their chances of correctly answering related questions and thus increased their chance of success in the whole test [63]; and predicting students' exam marks by using a decision tree [52].

The present research uses the decision tree, which is a popular classification technique. A decision tree induction algorithm consists of a greedy algorithm based on the divide and conquer philosophy that constructs a decision tree in a top-down recursive approach. Some of the qualities that create particular decision tree algorithms, such as the widely-used C4.5 algorithm, include the superior stability between precision, speed and interpretability of results. The C4.5 algorithm was invented by J. Ross Quinlan, who had previously invented the ID3 algorithm, the C4.5 predecessor, during the late 1970s and early 1980s. Since the invention of the C4.5 algorithm, it has been considered as a benchmark for many newer classification algorithms [32].

The present study uses the J48 decision tree algorithm that is an Open Source Java implementation of the C4.5 algorithm available in the Weka data mining package. The objective in using the algorithm is to build a model that can be used to predict the exam marks of new students. To build the model, a summary table generated from both the Reflect system and the student assessment dataset is used. A further discussion of the J48 implementation and the dataset is presented in Section 4.4.2.

2.6.3 Association Rules Mining

The purpose of association rule mining applied to educational data is to reveal relationships among the attributes in the data. These relationships among attributes and values are often represented by if-then relationships. An association rule mining algorithm is usually intended to find a combination of items that typically occur together in the data and/or a sequence of items that occur frequently in the database.

Although the application of association rules in e-learning posses a number of drawbacks [28] and measuring its interestingness is difficult and challenging [54, 87], there is a relatively large amount of research in EDM that employs association rules to tackle a number of tasks including: finding interesting association rules so that teachers may improve the performance of a Web-based Educational Adaptive Hypermedia System [72]; helping to measure the interestingness of association rules by using the cosine and added value measures [54]; discovering prediction rules in the AHA! course [71]; finding students' mistakes that often occur together [51, 53, 52]; and searching for important aspects of students' behaviour when working in a group [65].

In order to find any interesting relationships in the dataset used in the present study, a standard Apriori algorithm available in the Weka data mining package is implemented. Apriori is an algorithm that in a first step builds frequent item-sets and extract association rules from these frequent item-sets. The algorithm was first introduced by Agrawal and Srikant in 1994 [1] and since then has become a benchmark for association rule mining research. One of the advantages of the association rules mining technique is that it can predict any attribute, not just the class attribute. It also has the capability to predict the combination of attributes [89].

The Apriori algorithm can generate a lot of rules in the form of *IF-THEN* relationships. However, not all the rules generated by the algorithm are useful or *"interesting"*. Therefore, there are certain requirements that can be used to measure which rules are interesting. The commonly used tools to extract the rules are *support* and *confidence.* The support (coverage) is the number of instances that the algorithm predicted correctly, while the confidence (accuracy) is the number of instances that the algorithm predicted correctly as the proportion of the whole instances of the predicted attribute [89]. In this study, these two measurements of the rules interestingness are used to select the rules that were considered interesting. The main objective is to find any interesting relationships in the dataset that may be useful for teachers and students. A more detailed discussion of the dataset and how the Apriori algorithm was used is presented in Section 4.4.2.

2.7 Research Gaps

After reviewing a number of studies related to the use of data mining techniques to explore educational data, we were able to identify some gaps that can be explored and some differences that can be compared to our research. This section discusses the differences and gaps between the extant literature and the present study.

2.7.1 Self-Assessment Concept

Based on the review of the literature presented in Chapter 2, it seen that most EDM research that discussed students' self-assessment defined "self-assessment" from the number of student activities in either a learning management system (LMS), intelligent tutorial system (ITS) or online collaborative tool; for example, how many times students have done quizzes and assignments, sent messages to the forums, and used chat facilities. Self-assessment is rarely defined in the literature as a "direct act" by students in evaluating themselves, for example by rating themselves as "Well", "OK" or "NOT OK" in an LMS, ITS or collaborative tool.

In the Reflect system, the students are able to assess themselves by rating their understanding of a concept as "Very Well", "Well", "OK" or "Not at all" (more detailed information about this self-assessment is provided in Section 3.3). Each time students rate themselves in the Reflect system, a self-assessment task is recorded in the system. The total self-assessment is the sum of the students' self-assessment tasks performed in the Reflect system.

In regard to this process, we argue that the self-assessment concept in the Reflect system is different from the self-assessment concept adopted in the current EDM research (as discussed in Section 2.4). The self-assessment performed in Reflect is "direct", which means the students rate themselves directly in the system, while the current research concept of self-assessment is "indirect". In most research, the self-assessments were calculated and interpreted by researchers from the number of activities done in an educational system such as Moodle.

In this research, direct self-assessment data is used along with other student activity data gathered from the Reflect system. Therefore it is more closely aligned with students' perception rather than with system-inferred assessment of student learning.

2.7.2 System and Data

The majority of the current research in EDM utilised data from an LMS such as Moodle [42, 74, 87], an ITS such as SQL-Tutor [47] and ASSISTments [88] or an online collaboration tool such as Trac [64, 65]. These systems can record the activities performed by students through the system such as submitting quizzes and assignments, viewing tasks, joining discussions and posting messages into forums, but are not specifically designed for students to perform "self-assessment" tasks in which they can rate their understanding of a subject. As a result, past data mining experiments have been performed utilising the students' activity data and not the students' self-assessment data.

The Reflect system is different from the majority of learning management or intelligent tutoring systems. The Reflect system is different because it was designed to promote learner reflection and student self-assessment through a scrutable learner model. For example, in the Reflect system, students are able to compare their self-assessment in example solutions with the tutors' assessment. They can also view their progress on a graph that shows how close they are getting to the tutors' assessment. The students are also able to submit their solutions to the task that would be recorded as evidence. These facilities are usually not found in the majority of learning management or intelligent tutoring systems. In this regard, the present research uses complete data attributes that utilise not only the students' activity data but also the real students' self-assessment data. Indeed, although LMS can be setup so that students self-evaluate their skills, this functionality is embedded at the core of the Reflect system: it enables students to perform their self-assessment directly through the system, linking these to the tasks and learning objectives.

2.7.3 Methodology

Our research approach presented in Chapter 4 follows some of the methods proposed in the literature. For example, the present research follows a pre-processing process introduced in [75] to convert the class label into nominal attributes that can be used for the decision tree classification with the J48 algorithm. However, we then performed the attribute selection process using the Gain Ratio attribute evaluator function from the Weka data mining package. The attribute selection process is intended to select the most useful attributes used in the classification analyses. The classification analyses are then executed in Weka instead of Keel as in the previous research. The final step conducted in the experiments is to measure the accuracy of the model. The accuracy measurement for each experiment is presented in this thesis. This step was not presented in [75]; hence, there is no information about the accuracy of the model reported in that paper. As a result, the model can not be validated.

2.8 Conclusion

This chapter reviewed the relevant existing work in the educational data mining area. It reviewed some prominent related work in EDM research area. The literature review provided guidelines for the work carried out in this thesis. It identified the positive aspects of the previously published works that can be followed and potential problems that may be addressed in this thesis. For example, the steps in data pre-processing and discretisation are adapted from the previous work by Romero et al. [75]]. The chapter concluded with a discussion of the research gaps that are explored in the present study.

Chapter 3

Learning by Reflection in the Reflect system

Since the data used in the experiments comes from the Reflect system, we briefly present an overview of the system and its features. In this chapter, we also describe the process that students follow to do self-assessment in the Reflect system. The goal of the discussion in this chapter is to demonstrate and verify that the Reflect system supports students to learn by reflecting on their learning habits and experiences.

3.1 Introduction

As discussed in chapter 1, self-assessment helps learners to recognise their strengths and weaknesses, hence enabling them to focus on the areas that need improvement. This statement has been support by a large number of studies (i.e. [14, 77, 78]). Furthermore, Kay et al. [36] suggested that there are two key elements of student self-assessment: (1) identification of criteria and standards to be applied to their own work, and (2) evaluation of their work compared to the criteria and standards that have been identified in point 1. In the Reflect system, the two key elements of self-assessment are performed by the students through the following tasks:

1) Students read a task and assess the example solutions made available in the system.

2) Students provide their own solution to solve the problem

3) Student self-assess those solutions compared to the criteria the teachers have

defined for the task.

These three tasks are discussed further in Section 3.3. By performing these procedures, the students practise self-assessing themselves and their state of knowledge regarding a particular task. Thus, they have carried out learning by reflecting on their learning experiences.

3.2 Overview of the Reflect System

The Reflect system is a web-based educational teaching system used for learning, practising and testing programming knowledge. The system is not intended to be the sole means for learning a topic, but serves as a learning medium in addition to face-to-face teaching. The Reflect system is unique because it aims to promote learner reflection and student self-assessment through a scrutable learner model. This model shows the learners their progress [36]. The Reflect system offers extra attributes that are not commonly found in other learning management systems or intelligent tutoring systems, such as the student assessment of example solutions and comparison with the tutor's assessment. Reflect also allows students to view their progress on a graph, showing how close they are getting to the tutor's assessment. The section "What makes this site so special?" in Figure 3.1 presents the list of features that the Reflect system can offer.

3.3 Procedures of Self Assessment in Reflect

Students are instructed to use the Reflect system as an alternative resource for learning and practising their C programming skills. They are encouraged to work independently in a laboratory classroom or somewhere else with their personal laptops. They are given the freedom to practise solving the problems in the Reflect system at their own pace. There is no time limit for each task; however, students are encouraged to practise the tasks in the Reflect system according to the designated topic materials for each week.

In Reflect, students are expected to complete three stages of self-assessment. Each of these stages is described in the following subsections.

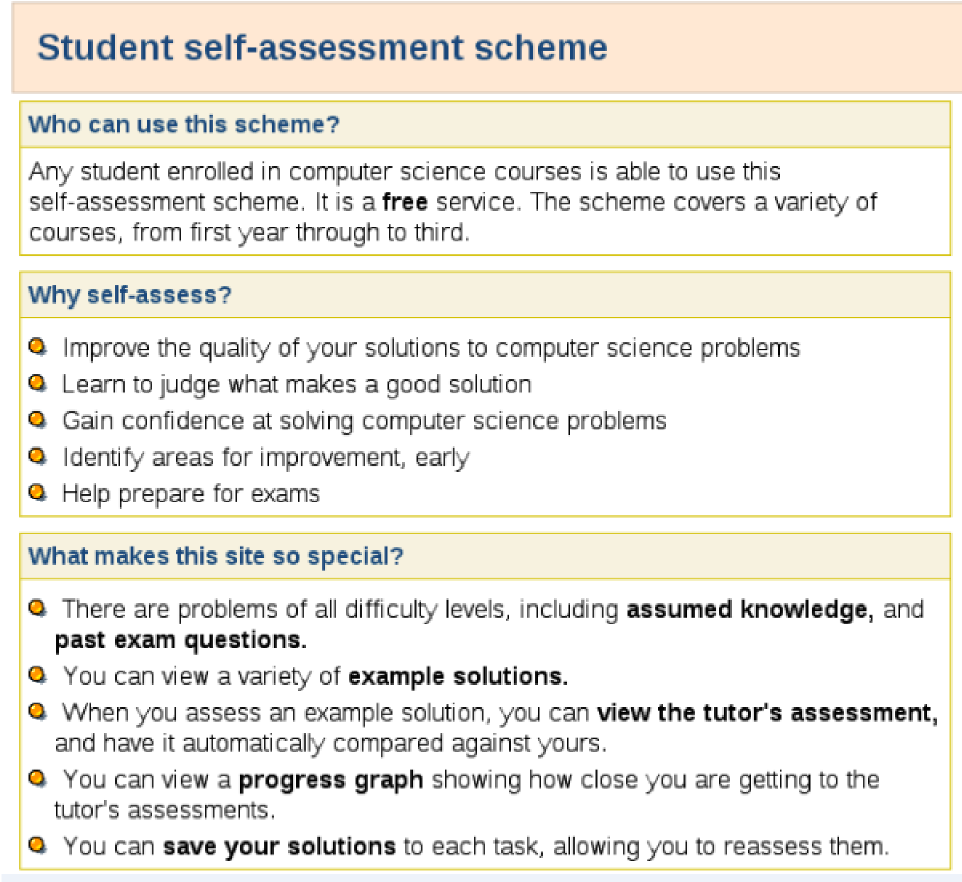


Figure 3.1: Features in the Reflect system

3.3.1 First Stage

After logging in, students are presented with a list of tasks that is mapped to one or more learning objectives (Figure 3.2). There are 24 tasks in the list for the students to practise. Students must select a task; after this they have to rate their understanding of the learning objectives related to the task. When reading the task and providing their sample solutions, the most important thing that students need to do is to evaluate the learning objectives of each task. Each criterion provided in the task corresponds to one of the learning objectives of the task. For each task, the learning objectives are designed to improve the students' understanding of the task.

The class teacher defines the mapping between the criteria and learning objectives that are related to the task.

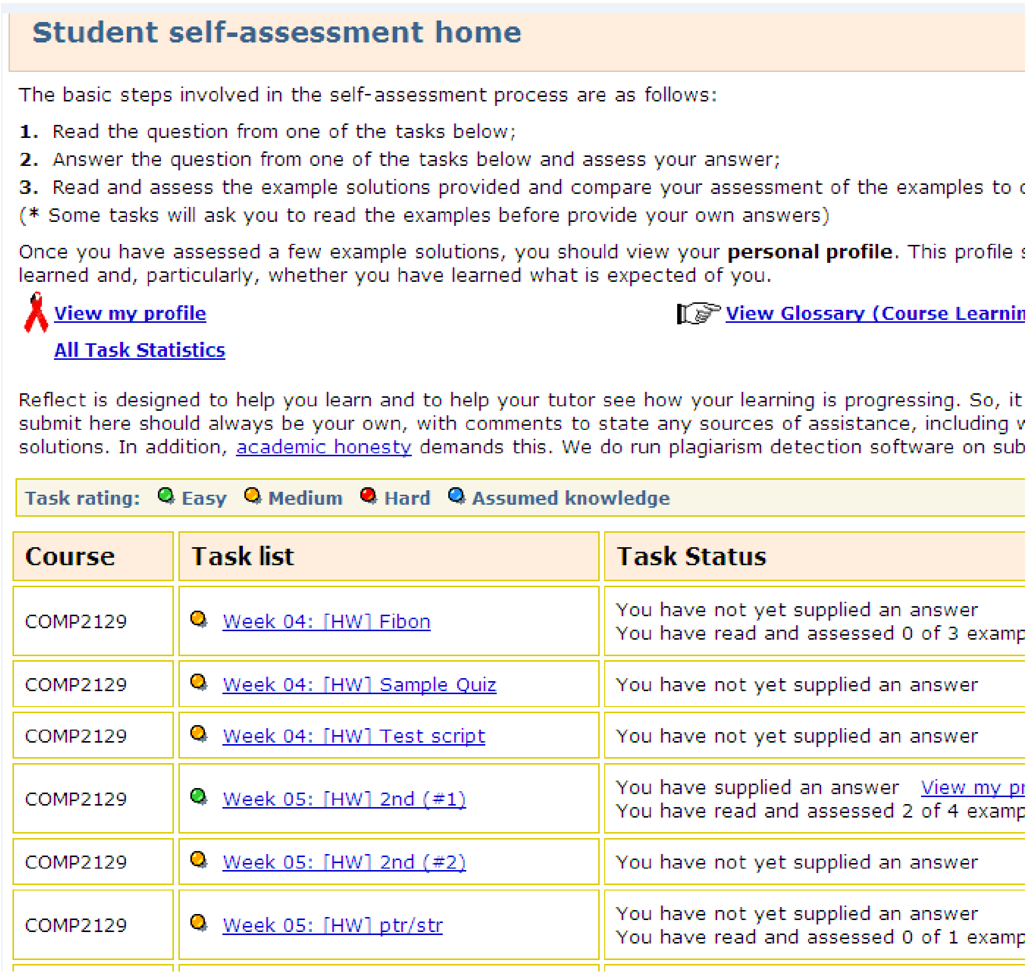


Figure 3.2: List of tasks presentation in the Reflect system

For example, in Figure 3.3, Task 12 (named "Week 04:[HW] Sample Quiz") has four related learning objectives (Figure 3.4), namely: (1) Similar concept in both C and Java - Control Flow, (2) Coding Style - Good Use of Indentation, (3) Coding Style - Useful Identifier Names, (4) Coding Style - Comment. Students rate themselves for each learning objective associated with the task according to their belief of how well they understand each concept. They rate their knowledge on a four point scales: *Very well*, *Well*, *OK*, and *Not at all*. The students who were confident that they understood well the learning objectives might choose Very well or Well options. On the other hand, students might choose the OK or Not at all options if they are not very confident that they understand the learning objective very well.

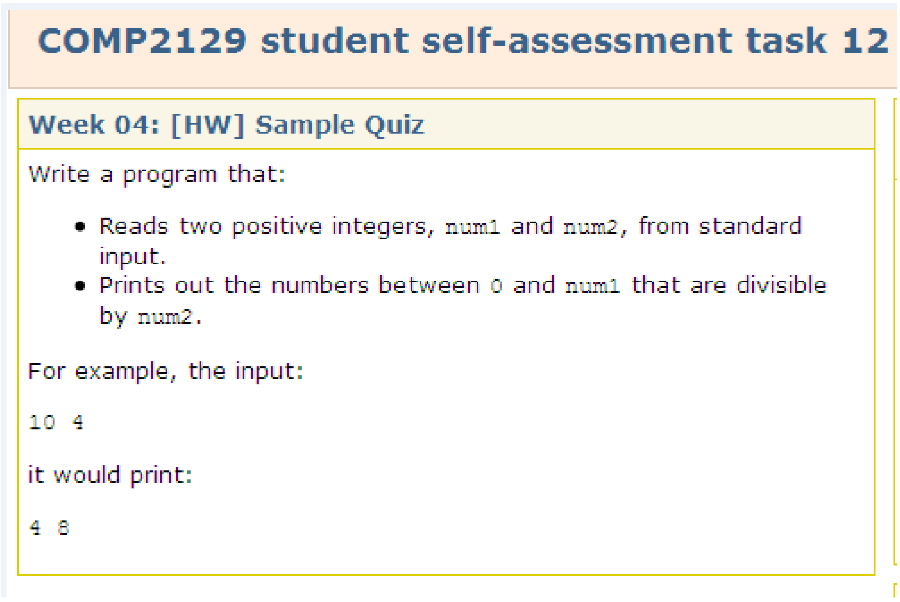


Figure 3.3: Student self-assess page in the Reflect system

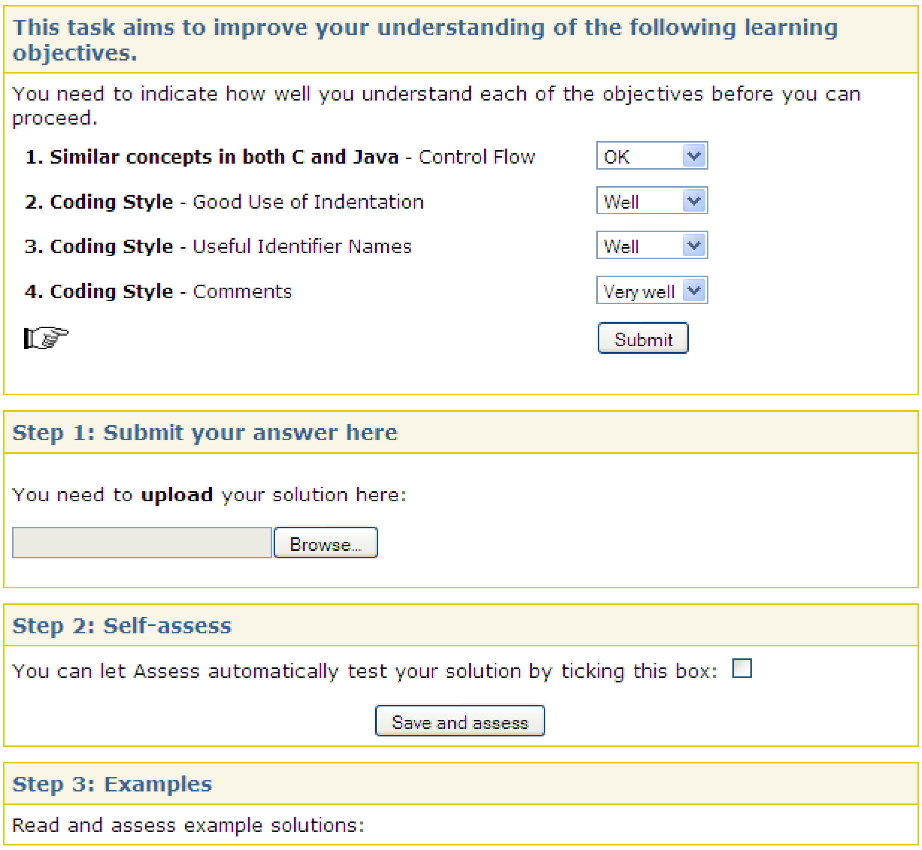


Figure 3.4: Student self-assess page in the Reflect system

The students' ratings (assessments) of learning objectives are recorded in the Reflect system as the self-assessment evidence for each student. The total self-assessment is the aggregation of the self-assessment tasks that consists of the total number students' assessments (rating) of their learning objectives (Very well, Well, Ok and Not at all) for each task in the Reflect system. For example in Figure 3.4, there are four learning objectives of task 12 (Figure 3.3) that needed to be rated by students. The rating is to indicate how well the students understand the learning objectives 1 to 4. For instance if the student feel that he is not very knowledgeable on the learning objective 1, he can rate this learning objective as "OK". But if he can write very good comments for the code then he can rate "Very well" on his understanding of this learning objective (learning objective 4).

If he rates only two learning objectives, it means that his total number of learning objectives (n_lo) for this task is 2. This means that he has done a partial task because he is not completed all four learning objectives. This partial task is also taken into consideration when calculating the number of task done (n_task). The total number of learning objectives for every student is calculated the based on the aggregation of all of learning objectives done by the student in every task. While the total number of tasks is calculated based on the number of task attempted either it is completed or partially completed. The attributes used for the experiments and their ranges are presented in 4.11.

3.3.2 Second Stage

Once the students have finished the above steps, they can submit their solution to the task. After they have submitted their solution they can self assess it against a set of marking criteria for the task as defined by the teacher. An example of a marking scheme page is shown in Figure 3.5. The students self assess their answers against each marking criteria by choosing one of the following options: *true*, *false* or *no opinion*. Their self assessment is saved in the system and can be revisited at any time in the future.

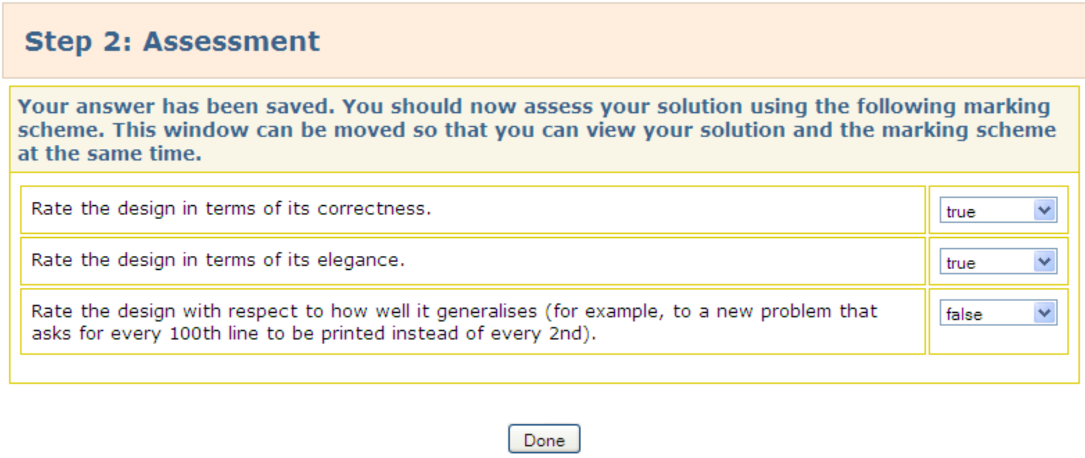
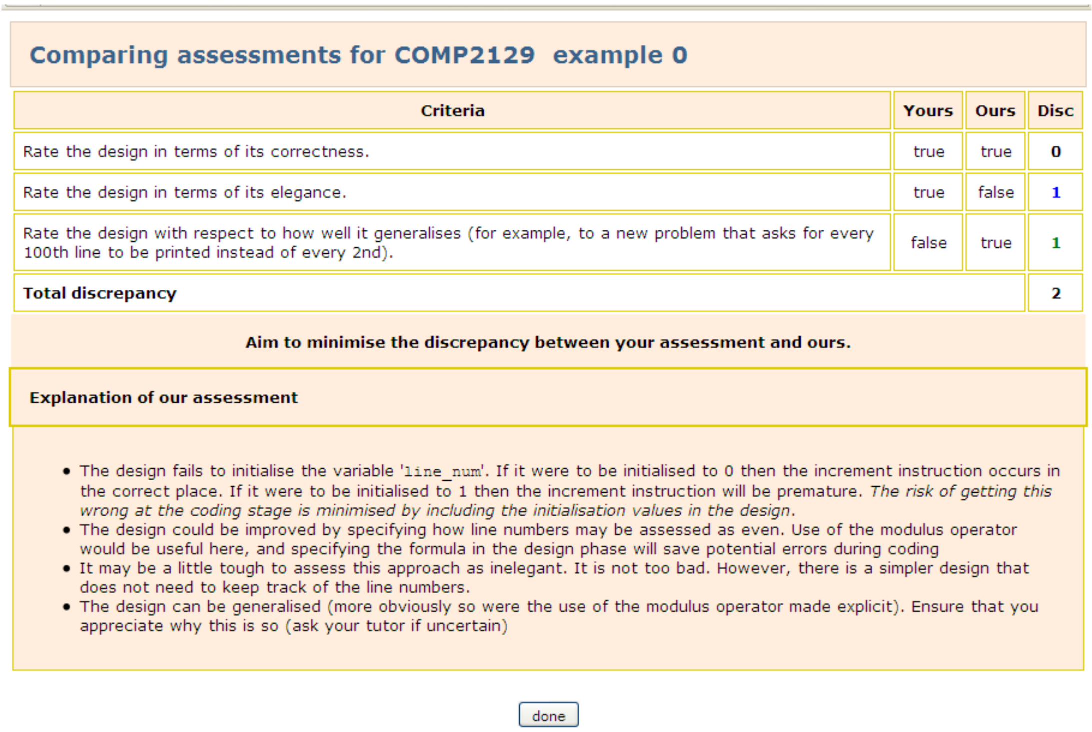


Figure 3.5: Self-assess marking scheme page

3.3.3 Third Stage

In the next stage, the students view example solutions as shown in Figure 3.7 and assess them according to a certain marking scheme. The example solutions have been pre-assessed by the teacher using the same marking criteria. The students' assessments are then compared to the teacher's assessment. Any discrepancies between the students' and teacher's assessment can be viewed by the students as shown in Figure 3.6.



Comparing assessments for COMP2129 example 0

Criteria	Yours	Ours	Disc
Rate the design in terms of its correctness.	true	true	0
Rate the design in terms of its elegance.	true	false	1
Rate the design with respect to how well it generalises (for example, to a new problem that asks for every 100th line to be printed instead of every 2nd).	false	true	1
Total discrepancy			2

Aim to minimise the discrepancy between your assessment and ours.

Explanation of our assessment

- The design fails to initialise the variable 'line_num'. If it were to be initialised to 0 then the increment instruction occurs in the correct place. If it were to be initialised to 1 then the increment instruction will be premature. *The risk of getting this wrong at the coding stage is minimised by including the initialisation values in the design.*
- The design could be improved by specifying how line numbers may be assessed as even. Use of the modulus operator would be useful here, and specifying the formula in the design phase will save potential errors during coding
- It may be a little tough to assess this approach as inelegant. It is not too bad. However, there is a simpler design that does not need to keep track of the line numbers.
- The design can be generalised (more obviously so were the use of the modulus operator made explicit). Ensure that you appreciate why this is so (ask your tutor if uncertain)

done

Figure 3.6: Discrepancy page in the Reflect system

3.4 Types of Evidence Recorded in Reflect

The evidence tags record information about the proof, that is, the solutions submitted by students during the self-assessment sessions in the Reflect system. There are two types of evidence that the system records based on the source of the gathered evidence: 1) evidence that was generated from a *Results from Local Test* (*RFLT* evidence) also called *student submission evidence*, and 2). *Example evidence* that was generated by the system when a student self-assessing an example solution. The example solution page is shown in Figure 3.7.

3.4.1 Student Submission (RFLT evidence)

A *student submission* evidence is a proof based on submission code submitted by a student and the code must pass the test set-up by the tutors in the system. The student submission evidence has only one value, namely, “1”. This is because the evidence is only recorded if the code submitted by a student passed the test. Thus, the evidence value of 1 indicates that student's submission code passed the test against a set of criteria set by the teacher in the system. No evidence value is recorded in the database if the code submitted did not pass the test.

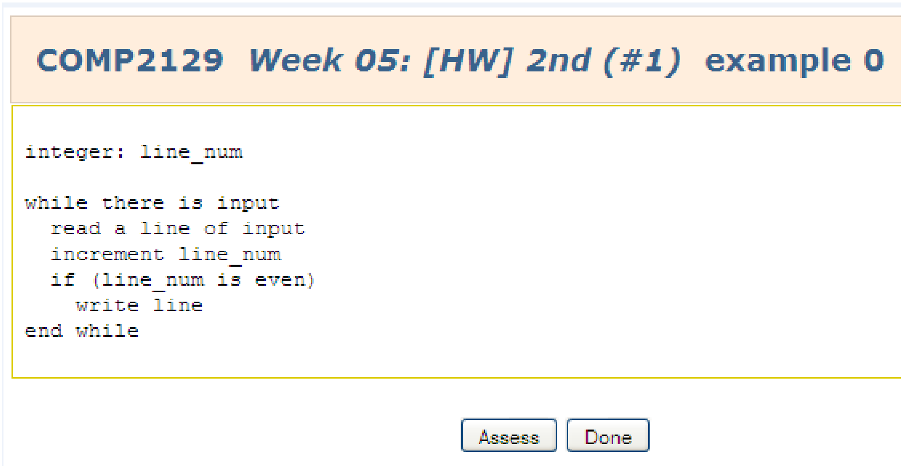


Figure 3.7: Example solution page in the Reflect System

3.4.2 Example Solution Evidence

An Example's evidence is a proof that is produced when a student self assesses the an example solution provided by the system. Example evidence is represented by an integer or a fraction and has a value from 0.0 to 1.0. Evidence values are calculated based on the discrepancy between the student's assessment and the tutor's assessment for a particular example as explained in Section 3.3. An evidence value of 1.0 indicates that the student's assessment for a particular example is the same as the tutor's assessment; thus, there is no discrepancy between the student's answers and the tutor's answers. On the other hand, an evidence value of 0.0 indicates that the student's answers are different to the tutor's answers thus the discrepancy score is the maximum. An example evidence value could also be a fraction such as 0.5. In this case, the 0.5 evidence value indicates that the student's assessment was similar to the tutor's assessment (this only occurs when the task has more than two options to select from). The smaller discrepancy value indicates that a student had a better understanding of the programming knowledge in that particular example of the problem. The students can view the discrepancy for each example solution that they assess in the Reflect system. This discrepancy page is shown in Figure 3.6

3.5 Conclusion

Chapter 3 shows how the Reflect system supports learning by reflection. This chapter demonstrates that users (in this case students) of the Reflect system can perform self-assessment by following the procedures as described in Section 3.3. These procedures include the rating of their understanding of each learning objective related to the task, providing answers to the tasks, self-assessing their answers and sample solutions against the criteria defined by the teacher and comparing the discrepancies between their answers and the teacher's assessment. As discussed in this chapter,the Reflect system provides users with a self-assess mechanism to helps them in learning.

Chapter 4

Research Approach and Methodology

This chapter describes the research approach and methodology of the present study. We begin by presenting an important discussion on the research questions, explaining how the study objectives presented in Chapter 1 were converted into a more focused research question and subsidiary research question. It is then shown how the subsidiary research question was translated into a number of hypotheses to be proved or unproved. This is followed by a discussion on the subjects and the types of data used in this study. Next, we present a discussion on how the data were prepared for mining. At the end of the chapter, a discussion on the EDM methods and algorithms used in the experiments is presented.

4.1 Research Questions and Hypotheses

As set out in Chapter 1, the main objectives of this study are to explore how data from an online learning system that supports learner reflection can be examined using EDM methods and to evaluate the effectiveness of these methods in: (1) extracting knowledge about the impacts of reflection on learning, (2) gaining knowledge about students' learning behaviour, and (3) identifying which behavioural patterns lead to positive or negative outcomes.

Based on these study objectives, the research question and subsidiary research question have been defined and presented in Section 1.3. We defined the primary research question as follows:

Can mining educational data help to better understand the impact of reflection on learning?

The primary research question was translated into a more specific subsidiary research question as follows:

Do students learn better by using the Reflect system?

In order to address the research question and subsidiary research question, we have conducted a study on students' learning behaviours when learning with a system that support learner reflection (i.e., Reflect). The present study employed EDM methods to examine the impact of those behaviours on the students' performance.

In addition, a number of hypotheses have been developed to address the research question and subsidiary research question. The main reason for translating the subsidiary research question into hypotheses is because a hypothesis can be directly tested; hence, its results can be measured and proved or unproved.

There have been five hypotheses developed. Hypothesis 1 to 3 (H1 to H3) were evaluated by using the Reflect dataset to examine the impact of the students' learning behaviours toward the students' exam performance. The main hypotheses (H1 to H3) are related directly to the use of Reflect system as a mean of reflection. The other two hypotheses (H4 and H5) were developed and tested by using the student assessment data from the course coordinator. The hypotheses that were developed to address the research question and subsidiary research question are discussed in this section.

First Hypothesis

Our first hypothesis is that the students who were able to submit RFLT evidence above the class average in the Reflect's student submission category (introduced in Section 3.4) were likely to be more successful and would perform better in the final exam. There are two submission categories in the Reflect system. In one category, students view example solutions and self-assess their answer against certain criteria set up by the teacher. Every time students self-assess an example solution, the Reflect system stores the assessment as evidence for self-assessment in this category. This is called the example solution evidence. The other category is a student submission category, also known as the RFLT evidence, in which students are required to write and submit their own solutions to solve a task category. Evidence is recorded every time a student submits a correct solution for the task. It is important to note that, in this case, the Reflect system records only the passed submission of RFLT evidence. In other words, an incorrect code submission would not be recorded by the Reflect system. In this regard, we hypothesised that the students who were able to submit the correct solutions (in the student submission category) at a rate above the class average were likely to be good students and hence would perform well in the final exam. Then, our first hypothesis was formed as:

Hypothesis 1: *Students who submitted RFLT evidence above the class average in the Reflect system are more likely to perform well in the exam.*

Second Hypothesis

Our second hypothesis related to the amount of time students spent on practising and solving tasks in the Reflect system. Students were expected to allocate a considerable amount of time learning and practising their programming skills in Reflect. By doing this, they would understand more of the core concepts of the programming they were learning. We believed that the more tasks students solve in Reflect, the more they learnt about the topic and this would be indicated by good performance in the exam. Therefore, our second hypothesis was formed as:

Hypothesis 2: *The total number of self-assessments performed in the Reflect system correlates positively to exam performance.*

Third Hypothesis

The third hypothesis was designed to identify and compare the relationships between the students who were motivated in practising the example solutions in Reflect and those who were not, and the benefits of those practices toward their knowledge gains when they had to submit their own solutions as student submission evidence. We assumed that students should practise example solutions as much as possible; hence, it would help them to solve more challenging questions in the student submission category. In this case, the challenges arose when the students were required to write, evaluate and test the code by themselves. Therefore, the third hypothesis was formed as:

Hypothesis 3: *Fewer submissions of the example solutions and RFLT evidence in the Reflect system would lead to poor exam performance*

In addition, this thesis also applies the data mining techniques to the students' assessment data. These are the data related to the students' performance in their lecture quizzes, homework/lab exercises and practical exams. The data were obtained from the course coordinator who taught the subject. In this regard, the purpose is to investigate the impacts of these academic practices on students' exam marks. This additional focus of the research may or may not directly contribute to the understanding of the impact of reflection on learning but it is important to learn about students' performance in general. For example, by knowing early at the beginning of the semester that some students have not made good progress in learning the topics being taught, the teacher could offer an extra tutorial or take other remedial actions. With regard to this additional research, the following two hypotheses were developed:

Fourth Hypothesis

Hypothesis 4: *Students who attend and achieve good marks on all quizzes have a better chance at performing well in final exam.*

This hypothesis was formulated to investigate the correlation between the quiz results and exam performance. We were interested to know if quiz performance and attendance had significant effects on student readiness for the final exam.

Fifth Hypothesis

Hypothesis 5: *Students who miss weekly homework/lab exercises are at an increased of risk of performing poorly in the final exam.*

Lab practices are the most important learning component in most programming courses. It is even more important for students to attend a lab session if it is designated for a practical exam as were for the tree lab sessions in the Software Construction I course. For these specific weeks, students could experience the real exam situation in which they could practise solving questions similar to those that would be presented in their exam, under the supervision of a teacher. Hence, we hypothesised that missing even some of the weekly lab sessions would be a real disadvantage for the students.

Examining these hypotheses will provide answers to the subsidiary research question and in turn they together will provide an answer to the primary research question. Next, before we discuss the present study's methodology, we describe the subjects and source of data used in the study.

4.2 Subjects and the Data

4.2.1 Subjects and Course Context

The subjects used in this research were students enrolled in the course entitled "Software Construction I" (SOFT2130/2830) at the School of Information Technology, the University of Sydney, in the second semester of 2007. In this course, students learn how to program in the C programming language. In 2007, 175 students were enrolled in the course, with 14 categorised as advanced students. In regard to the course assessment policy, all students except the advanced students had three assessment components: 1) weekly homework/labs, 2) weekly tutorial quizzes, and 3) a final exam. For advanced students, there was an extra assessment component consisting of an assignment. Among the 175 students, 156 (89.1%) of them completed the course at the end of the semester. Of the 156 students who completed the course, 109 (69.9%) students passed and 47 (30.1%) failed the course. The final grade distribution of the course is shown in Table 4.1.

Table 4.1: Distribution of student final grade

Grade	Label	Percentages (%)	Distribution
High Distinction	HD	6.4	7 out of 109
Distinction plus	D	24.8	27 out of 109
Credit plus	CR	56.0	61 out of 109
Passes	P	12.8	14 out of 109
Fail	F	30.1	47 out of 156

4.2.2 Data Source

The datasets used in this research came from two sources: 1) data about students' activities gathered from the Reflect system (as explained further below); and 2) data about students' assessment obtained from the course coordinator who taught the course. In conducting the experiments, both datasets were used. The following subsections discuss each dataset.

Students' Activity Dataset

Data about the students' activities were gathered from the Reflect system, which is an Intelligent Tutoring System that promotes learning by reflection. An intelligent tutoring system (ITS) is a computer system that aims to provide immediate and customized instruction or feedback to learners, usually without intervention from a human teacher [66]. Based on this definition the Reflect system can be categorized into an Intelligent Tutoring System (ITS) since it has capability to immediately give feedback to users regarding their understanding of programming concepts. Reflect also capable to evaluate programming codes submitted by students by comparing them with the pre-programmed codes from tutors. The Reflect system is provided by the university for the students to use as an alternative source of learning and practice in the Software Construction course. The data were gathered from the Reflect database in the XML format. An example of this data is illustrated in Figure 4.1.

```
<?xml version="1.0" ?>
<reflect_data>
  <tasks>
    <task published="True" task_id="SOFT2130/Task11">
      <task_name>
        Week 05: [HW] 2nd (#2)
      </task_name>
      <learning_objectives>
        <learning_objective>
          SOFT2130/Core/Similar concepts in both C and Java/Control Flow
        </learning_objective>
        <learning_objective>
          SOFT2130/Core/Similar concepts in both C and Java/EOF for detecting the end of input
        </learning_objective>
        <learning_objective>
          SOFT2130/Core/Coding Style/Good Use of Indentation
        </learning_objective>
        <learning_objective>
          SOFT2130/Core/Coding Style/Useful Identifier Names
        </learning_objective>
        <learning_objective>
          SOFT2130/Core/Coding Style/Comments
        </learning_objective>
      </learning_objectives>
    </task>
    <task published="True" task_id="SOFT2130/Task12">
      <task_name>
        Week 04: [HW] Sample Quiz
      </task_name>
      <learning_objectives>
        <learning_objective>
          SOFT2130/Core/Similar concepts in both C and Java/Control Flow
```

Figure 4.1: A screen shot of Reflect data in XML format

The student activity data are organised in a hierarchical structure wherein each task consists of a task name and a number of learning objectives. The number of learning objectives for each task varies between one and five. A learning objective can appear in one or more tasks.

In general, the XML data can be viewed as two parts: 1) one section contains information related to a task, 2) one section is organised according to the student ID (not shown in Figure 4.1). The following is an example of the first section of the data.

As seen in the above list, task 11 is enclosed by two <task> tags. The task_id attribute identifies the task number. For example, in Figure 4.1 the task_id="SOFT2130 /Task11" is followed by five learning objectives that belong to that task. In addition to the task number, the task name consists of the allocated week for the task, the type of assessment ([HW] means homework/lab) and the task name itself; for example, 2nd (#2) is the name of the task asking for the printing of every second element. The learning objectives are presented in the learning objective tags:

The second part of the XML data is structured according to the *student ID*. The data for each student, identified by student id, are followed by two other important tags, namely, the *self-assessment* and *evidence tags*. The self-assessment tags record the self-assessment values ("Very well", "Well", "OK" and "Not at all") for each learning objective related to the task. Each self-assessment value is recorded as self-assessment evidence in the students' learning model each time students rate (self-assess) themselves as described in Section 3.3.1. This self-assessment evidence is different to the "results from local tests" (RFLT) that was discussed in Section 3.3. The student self-assessment data are presented in the following format:

Meanwhile, the XML data related to the evidence were recorded in Reflect system as follows:

- Evidence generated from Results From Local Tests (RFLT evidence)

- Evidence generated from self-assess an example solution

Furthermore, the students' activity data consist of a number of attributes as summarised in Table 4.2.

Students' Assessment Dataset

The students' assessment data were spread over several worksheets in Microsoft Excel. These include lecture quiz marks, homework and lab marks (including practical marks on weeks 4, 8 and 12), quiz marks, and the final exam marks of students enrolled in the Software Construction course for the year 2007. The data were obtained from the lecturer who organised and taught the course. Table 4.3 summarises the attributes of the students' assessment data.

Table 4.2: List of attributes in Reflect dataset

Attribute	Description	Data Type
Example0	evidence recorded from students self-assess example solution 0	Numeric
Example1	evidence recorded from students self-assess example solution 1	Numeric
Example2	evidence recorded from students self-assess example solution 2	Numeric
Example3	evidence recorded from students self-assess example solution 3	Numeric
RFLT evd	evidence generated from student submission	Numeric
n_task	number of tasks done in Reflect	Numeric
n_lo	number of learning objective done in Reflect	Numeric
Tot_selfAssess	total self assessment done in Reflect	Numeric

Table 4.3: List of attributes in students' assessment dataset

Attribute name	Description	Data type
L2_mem_diagram	Week 2 lecture quiz mark	Numeric
L5_pointers	Week 5 lecture quiz mark	Numeric
L8_scope	Week 8 lecture quiz mark	Numeric
Tot_Lec_qz	Total lecture quiz marks (%)	Percentage
Lec_qz_attd	Total lecture quiz attendances (%)	Percentage
HWLwk4	Homework/Lab (prac exam) week 4	Numeric
HWLwk8	Homework/Lab (prac exam) week 8	Numeric
HWLwk12	Homework/Lab (prac exam) week 12	Numeric
Tot_HWL	Total homework/lab marks (%)	Percentage
HWL_attd	Total homework/lab attendances (%)	Percentage
Tot_qz	Total quiz marks (%)	Percentage
qz_attd	Total quiz attendances (%)	Percentage
exam	Final exam mark	Numeric

4.3 Preparing Data for Mining

The general process of data mining involves: gathering the data; pre-processing the data; applying data mining algorithms; and interpreting the results. Similar data mining steps were also applied to this thesis. However, as the Reflect data were not designed for data mining analyses, it was necessary to perform more manual data pre-processing and choose the data attributes carefully.

This section focuses on the data mining process as part of the overall research approach. It presents the methodology for mining the Reflect and student assessment data. The overall process of data mining is shown in the diagram in Figure 4.2. Each box in the diagram is discussed in the following subsections starting with the pre-processing stage. The discussion in this section is, however, limited to discussing the methodology, that is, how the experiments were conducted.

4.3.1 Data Pre-Processing

In a large database, it is commonplace to have data that are incomplete, inconsistent, noisy and contain errors [32]. The data pre-processing task consists of cleaning incomplete and inconsistent data, removing noises and errors, and integrating and transforming the data into an appropriate format for mining. The following section discusses the pre-processing techniques used in the present study.

Cleaning the Data

After the data were extracted from the Reflect database in XML format, the next stage was to clean them of inconsistencies, errors and noises before they were ready to be mined. We carried out the process of data cleansing manually by searching for incomplete data, noises and errors and removing them from the data. The errors in the data came from the data records of unintended or non-enrolled users. These users included a system administrator, a number of postgraduate students who used the system for testing, and other students who had not enrolled in the course but used the Reflect system for learning and practising to solve C programming programs. The errors also came from a user who had two user name log-ins. This could have happened when the student changed their user name log-in at some point during the semester. Noise in the data may have come from an outlier; for example, we found that there was a student who rated himself as "OK" 367 times for a learning objective. This rate was considered excessive and unrealistic because most of the students from the same class rated themselves as "OK" only 41 times on average.

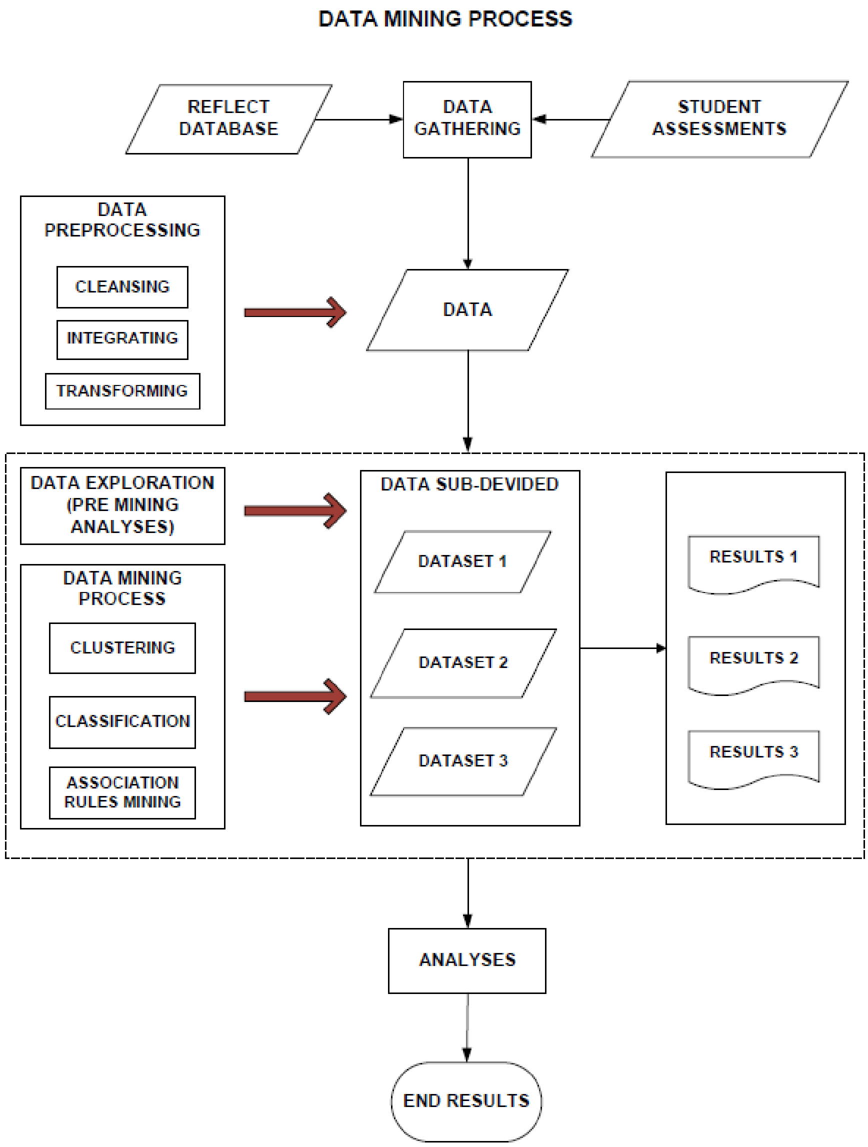


Figure 4.2: Diagram of the process of data mining

The unrealistic figure caused an imbalance in the data distribution and we indicated it as an outlier. In this particular case, our approach was to replace the outlier value with the class average value of 41. A summary of the types of errors found in the Reflect database is shown in Table 4.4.

Integrating and Transforming the Data

Data integration is a process of integrating or merging the data from multiple tables or databases into a coherent data format [32]. Data mining processes often involve

Table 4.4: Source of errors in the Reflect data

Sources of errors	Explanation
Record of DB admin	DB admin did not enrolled for the course
Double user names	Resulted in data redundancy
Postgraduate students	The course is for undergraduate
Other students	Undergraduate students who did not enrolled but using the Reflect system
Noises	Unrealistic data points way above or under class average

retrieving and analysing multiple data attributes that are spread across several tables or databases. As mentioned in Section 4.2.2, our data sources came from two sources: the students' activity data gathered from the Reflect database and students' assessment spreadsheet obtained from the course coordinator. Therefore, these data need to be summarised into a summarisation table consists of all necessary attributes for data mining experiments. For this purpose, a summarisation table was as shown in Table 4.5.

Data transformation is a process involving a number of techniques including *smoothing* to remove noise from the data; *aggregation* which is a summary operation applied to the dataset; *generalisation* in which the lower level data are replaced by higher-level concepts; and *normalisation* where the attributes data are scaled thus they fall within a small range (i.e., 0.0 to 1.0). [32].

In this research, the smoothing technique to remove noise in the Reflect dataset was employed as explained in the Section 4.3.1. The aggregation technique was used for example to summarise the weekly homework/lab marks and quiz marks into total homework/lab (Tot_HWL) marks and Total quiz (Tot_qz) marks. The normalisation technique was used to normalised the attributes in the clustering analysis.

The data also needed to be transformed into an appropriate format for mining. The common formats used for mining data with the Weka data mining system are the attribute-relation file format (ARFF) and comma-separated value (CSV) format. Therefore, the dataset was transformed into either ARFF or CSV format to be suitable for use in Weka. The ARFF and CSV format are easy to use and acceptable in the Weka system. Moreover, to increase the interpretation and comprehensibility of the data, it is sometimes necessary to perform data discretisation; that is, to transform the numerical value of a data attribute into a nominal or ordinal one [69].

Table 4.5: List of all attributes in the summary dataset

Attribute Name	Description
L2_mem_diagram	Week 2 lecture quiz mark
L5_pointers	Week 5 lecture quiz mark
L8_scope	Week 8 lecture quiz mark
Tot_Lec_qz	Total lecture quiz marks (%)
Lec_qz_attd	Total lecture quiz attendances (%)
HWLwk4	Homework/Lab (prac exam) week 4
HWLwk8	Homework/Lab (prac exam) week 8
HWLwk12	Homework/Lab (prac exam) week 12
Tot_HWL	Total homework/lab marks (%)
HWL_attd	Total homework/lab attendances (%)
Tot_qz	Total quiz marks (%)
qz_attd	Total quiz attendances (%)
Example0	Evidence recorded from students self-assess example solution 0
Example1	Evidence recorded from students self-assess example solution 1
Example2	Evidence recorded from students self-assess example solution 2
Example3	Evidence recorded from students self-assess example solution 3
RFLT_evd	Evidence generated from student submission
Tot_selfAssess	Number of self assessments done in Reflect
n_task	Number of tasks done in the Reflect
n_lo	Number of learning objectives done in the Reflect
exam	Final exam mark

4.3.2 Constructing the Datasets for Experiments

Choosing Correct Data Attributes

Attribute selection plays an important role in data mining analysis. Choosing incorrect data attributes can mislead the interpretation of the results, resulting in invalid results or results that are difficult to be validated. The following guidelines were followed in choosing the data attributes for our data mining analyses.

- Data attributes were chosen according the experiments conducted; that is, each experiment could use different data attributes. The experiments conducted were based on the research questions or the hypotheses that followed.

- For the purpose of classification analysis with J48, we used the attribute selection tools available in the Weka data mining package called the *Gain Ratio Attribute Evaluation.* Initially, we also used *Info Gain Attribute Evaluation* but after comparing preliminary results, there were not much different between the two; hence, we decided to used only the *Gain Ratio Attribute Evaluation.*

Constructing the Datasets

From the list of data attributes in Table 4.5, three types of datasets were constructed for testing the hypotheses: 1) a numerical dataset, 2) a numerical dataset with categorical class attributes, and 3) a nominal dataset. The datasets were used according to the type of algorithms executed. For example, the K-means algorithm accepts numerical as well as nominal attributes, the J48 algorithm takes numerical data but requires a nominal or ordinal class label, and the Apriori algorithm requires all attributes to be in nominal form. The following subsection presents overview of these datasets. In order to maintain anonymity of the subjects, pseudonyms are used to replace student's user name.

Numerical dataset

In a numerical dataset, all data attributes are in a numeric form and can contain any numeric attributes including percentages. This dataset is used for statistical and clustering analyses with K-means. Table 4.6 shows an example of data in this format.

Table 4.6: An example of data attributes in numerical form

UserID	Tot_HWL	Tot_qz	n_task	n_lo	exam
student01	1.00	1.00	22	24	93
student02	0.98	1.00	24	24	92
student03	1.00	1.00	22	24	92
student04	1.00	1.00	23	24	90
student05	0.75	0.70	16	14	65
student06	0.98	0.98	24	24	78
student07	0.35	0.50	6	7	48

Numerical dataset with Nominal Class

The J48 algorithm used in the experiments requires that the class attribute, in this case the exam marks, must be in a categorical or nominal form. Therefore, a dataset of numerical attributes with nominal class label was created. In this dataset, the exam marks were manually grouped into the following categories: *"Low"* if the exam mark is <50; *"Moderate"* if the exam mark >=50 and <75; and *"High"* if the exam mark >75. The lower bound 50 was chosen to follow one of the course requirements that requires students to achieve at least 50 in the final exam in order to pass the course. With this regard, the exam mark 50 is the minimum requirement to pass the Software Construction I course. The upper bound 75 was chosen to represent the top students who achieve *Distinction* or *High Distinction* for the final grade. The moderate range between 50 and 75 was a representation of the majority students in the class that filled the gap between the lower bound and the upper one.

Although there are algorithms that accept all numerical attributes such as regression tree, REPTree and linear regression, we argue that the categorisation of class labels can enhance the interpretation and comprehensibility of the results [75]. This is particularly more useful if the results related to the pedagogical information and will be presented to the educators who may not have data mining background. Therefore, the numerical dataset with a categorical class was used for classification analyses. Table 4.7 shows an example of this dataset. The last data attribute in the table is a class attribute that was converted into three categorical exam marks labelled as *"High"*, *"Moderate"*, and *"Low"*.

Table 4.7: An example of a numerical dataset with a categorical attribute

UserID	Tot_HWL	Tot_qz	n_task	n_lo	exam_cate
student01	1.00	1.00	22	24	High
student02	0.98	1.00	24	24	High
student03	1.00	1.00	22	24	High
student04	1.00	1.00	23	24	High
student05	0.75	0.70	16	14	Moderate
student06	0.98	0.98	24	24	High
student07	0.35	0.50	6	7	Low

Nominal Dataset

A nominal dataset was used for association rule mining analysis utilising Apriori algorithm. This dataset is also known as a discretised dataset wherein all the numerical data attributes were transformed into categorical values. Data discretisation can enhance the interpretation and comprehensibility of the dataset [75]. Table 4.8 shows an example of dataset in nominal form.

Table 4.8: An example of data attributes in categorical form

UserID	lab_ percent	quiz_percent	num_task	num_lo	exam_cate
student01	High	High	High	High	High
student02	High	High	High	High	High
student03	High	High	High	High	High
student04	High	High	High	High	High
student05	High	Medium	Medium	Medium	Moderate
student06	High	High	High	High	High
student07	Low	Medium	Low	Low	Low

4.4 Application of EDM Methods

This section describes the approaches that were followed to answer the research question and its subsidiary research question as discussed in the previous section. We describe the data exploration by means of statistical methods, followed by a discussion on the data mining techniques and algorithms used in the study. The results of the data mining experiments are presented and discussed in a separate chapter (Chapter 5).

4.4.1 Exploring Data by Statistical Analysis

Statistical analysis often provides a starting point for data exploration. In this study, a number of correlation and regression analyses were conducted to learn about relationships and to measure whether one data attribute is significantly correlated to others. The analysis was also intended to investigate whether the movement of a dependent variable was caused by one or more independent variables. For example, our main objective was to utilise statistical analysis to find the correlations between:

- the number of Rflt evidence submitted in the Reflect system and the final exam mark.
- the number of total self assessments performed in the Reflect system and the final exam mark.

Furthermore, statistical analyses were performed to analyses the relationships between the lecture quiz and homework/lab marks and the final exam mark. The analyses that are related to this were:

- the lecture quiz marks and the final exam mark.
- the homework/lab marks and and the final exam mark.

Besides those objectives, statistical analysis can also be used to determine the variable that best distinguishes one group from another.

4.4.2 Educational Data Mining Methods

Simple statistical analyses are usually quite limited in terms of the results provided. For example, in this study, simple analysis may not be able to distinguish between groups of weak students and strong ones. To gain more useful insights into the data, we therefore utilised more advanced data analyses, namely, data mining techniques.

In this study, we utilised the three most well-known data mining algorithms and techniques, namely, the K-means algorithm for clustering the groups of students, the C4.5 and J48 (decision tree) algorithm for classification, and the Apriori algorithm for association rule mining. While statistical analyses were conducted to find true relationships among the attributes, the educational data mining methods applied to the Reflect data were more important in order to answer the research question and the related hypotheses. This is because the EDM methods, such as clustering, classification and association rule mining, can offer more meaningful results, way beyond the statistical analyses.

Clustering Students

Our clustering experiments utilised a standard K-means clustering algorithm to cluster groups of students based on their similar behaviour in using the Reflect system. The

K-means clustering algorithm is considered as one of the most reliable and widely used clustering algorithm in the data mining research community [75]. Our objective is to distinguish the stronger group of students from the weaker group; that is, to identify the aspects in regard to learning characteristics when using the Reflect system that separate these groups of students. In other words, we wanted to study the learning characteristics that led to an increase in student performance.

A number of clustering experiments were conducted by using the numerical dataset summarised from the student activities performed in the Reflect system. A summary table with four attributes related to students' activities in the Reflect system was built. These attributes were chosen as they represented the activities by students in Reflect. The attributes (as shown in Table 4.9]) are the RFLT evidence introduced in Section 3.4.1, the number of tasks and the number of learning objectives done in Reflect, and the total number of self-assessments each student performed in Reflect. The last mentioned attribute is an aggregate of the total number of self-assessments conducted by the students for the example tasks 0 to 3 (i.e., example0, example1, example2 and example3 tasks) that were submitted in the Reflect system.

One of the issues we found when using the K-means algorithm is how to set the number of clusters (k). In this case, we had to run a number of experiments and select the number of k clusters manually. We chose a range of clusters from 3 (k=3) to 5 (k=5). From several repeated experiments, we found that the result of the K-means clustering with k=3 was more interesting than the others. The clusters resulting from the K-means clustering with k=3 showed a higher similarity between the objects within the clusters if we compared to the clusters resulting from the K-means clustering with k=4 or k=5. The similarities between objects were measured by using *Euclidean* distance option available in Weka data mining tool.

At this stage, our experiments showed that the main discriminant attribute that separated the clusters was the student submission (RFLT) evidence. In this regard, this attribute can be used as an indicator of how well a student performed in Reflect. As explained in Chapter 3 the student submission (RFLT) evidence can only be recorded in the system if it passed the set of criteria set up by the tutor. This means that the more student submission evidence recorded in the Reflect system, the better the student performance in Reflect. This finding is coherent with the finding in a previous study [35] that suggested the Reflect system provides motivated students with help and the ability to understand the programming code closest to the teachers' viewpoints.

Consequently, we assumed that this help would encourage students to use Reflect more often. Hence, as a result, students' skill and knowledge increased significantly. However, this fact was not similar for all students. Some weak students may have found the tasks in the Reflect system difficult and hard to solve. The results of the K-means clustering and a discussion of the findings are presented in Section 5.1.2.

Table 4.9: Set of attributes used in the K-means clustering

Attribute Name	Description
RFLT_evd	Evidence generated from student submission
Tot_selfAssess	The number of self assessments done in Reflect
n_task	The number of tasks done
n_lo	The number of learning objectives done

Classification

We utilised a simple yet powerful J48 algorithm for the classification analyses provided by Weka data mining package. This algorithm is also known as the *decision tree* algorithm. The J48 algorithm generates decision trees that can be used to extract classification rules. In this study, the objective was to classify students according to their exam marks. The classifier or called *model* resulted from the experiments can be used to predict the final exam mark of new students.

Since J48 algorithm requires that the class attribute, in this case the exam marks, must be in categorical or nominal form, a categorical attribute of exam marks named exam_cate was created as discussed in Section 4.3.2. The exam_cate comprises of three classes: high, moderate, and low. This categorisation is different to the categorisation of the final grade students received at the end of semester. The student's final grade was calculated to include the final exam marks and other assessment components including assignments, quizzes, homework and lab, and practical exam. The assessment components and the percentages that will contribute to the students' final grade are arranged by the course coordinator and agreed by the students at the beginning of the semester. The final student's grade is not the main focus of our study, hence it is not discussed in this thesis.

Classification with the Students' Activity data

In the first round, the experiments were conducted using dataset gathered from the Reflect system (students' activity data). The dataset consist of nine data attributes as described in Table 4.2. Initially, we selected all attributes in the Reflect dataset into the Weka pre-processing tool. In most classification analyses, however, not all attributes are equally useful for predicting the target. Therefore, there is a need to evaluate the usefulness of each attribute used in experiments. For this purpose, we employed an automatic attribute selection called the *Gain Ratio Attribute Evaluation* (GainRatioAttributeEval) available in Weka. The Gain Ratio attribute evaluator works with a search mechanism called *Ranker* to rank the attribute according to its usefulness. In this case, the ranker searches for and list the attributes according to *the best splitting* criteria. The best splitting criteria is a method that determines the attribute to first split the tree and ranks them in order as illustrated in Figure 4.3.

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 9 exam_cate):
        Gain Ratio feature evaluator

Ranked attributes:
 0.231    7 Rflt_evd
 0.208    8 Tot_selfAssess
 0.183    1 n_task
 0.182    2 n_lo
 0        3 Example0
 0        6 Example3
 0        4 Example1
 0        5 Example2

Selected attributes: 7,8,1,2,3,6,4,5 : 8
```

Figure 4.3: Ranking of the Reflect data attributes in the Gain Ratio

The Gain Ratio attribute evaluator rated Rflt_evd, Tot_selfAssess, n_task and n_lo attributes as the top four attributes in the list, while Example2, Example1, Example3 and Example0 was rated as four attributes with the lowest ranking in the list. This suggests that the variable Rflt_evd would be the first splitting attribute when a decision tree was created, followed by Tot_seflAssess, n_task and n_lo attribute respectively. This also indicates that the Rflt _evd and Tot_selfAssess attributes were good discriminants for the decision tree.

At this stage, the dataset were ready for mining. The experiments were conducted by choosing the default *10-folds cross-validation* test option. The 10-folds cross-validation test option works by splitting the training data into 10 approximately equal size of subsets named *folds.* Out of ten folds, nine were used for the training and the one remaining was used for testing. A new model was built from the training data and it was evaluated using the left-out fold (testing data). When the process was completed for the first test fold, a new fold was selected for testing and the remaining folds were used for training. The process was repeated until all folds had been used for testing. The system was set-up to verify that each instance in the full dataset was used only once for testing and it was used when it was not used for the training. In Weka, the class proportion is preserved when the data is divided into folds. This means each fold contains approximately the same number of instances for high, moderate and low for the attribute exam_cate. Therefore, the cross-validation method in Weka is called a *stratified* cross-validation. The stratified cross-validation delivers slightly improved performance estimates over un-stratified cross-validation [89].

In decision tree analysis, there are many ways to measure accuracy of a classifier (model). One common way is by examining how many instances were correctly classified into classes. In the Weka J48 analysis tool, the correctly classified instances are shown in percentage form. The accuracy of the models can be compared with the baseline accuracy. The baseline accuracy could be the prediction of the majority class from the dataset. The finding and discussion of the classification results by using the Reflect dataset will be discussed in Section 5.1.

Classification with Students' Assessment Data

In the second round, the experiments were conducted using students' assessment data obtained from the course coordinator who taught the course. The dataset, initially consist of 13 data attributes as described in Table 4.3. However, some data attributes were redundant, hence they could not be included. For example, the total lecture quizzes (Tot_Lec_qz) mark were not included because it was represented by individual lecture quiz marks (L2_mem_diagram, L5_pointer and L8_scope). Similarly, Tot_HWL attribute were preferred over HWLwk4, HWLwk8, HWLwk12 because it aggregated not only homework/lab marks in week 4, 8 and 12 but also all weekly homework/lab marks.

After the data attributes were selected, the Gain Ratio Attribute Evaluation (GainRatioAttributeEval) as introduced in the previous section were employed to examine the usefulness of each data attribute. The ranking of the attributes based on the Gain Ratio attribute evaluator is shown in Figure 4.4.

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 9 exam_cate):
        Gain Ratio feature evaluator

Ranked attributes:
 0.344    4 Tot_qz
 0.3147   2 Tot_HWL
 0.1347   6 L2_mem_diagram
 0.1169   1 Lec_qz_attd
 0.1088   8 L8_scope
 0.1041   5 qz_attd
 0.0968   3 HWL_attd
 0.0925   7 L5_pointers

Selected attributes: 4,2,6,1,8,5,3,7 : 8
```

Figure 4.4: Ranking of the students' assessment data attribute in the Gain Ratio

From the Gain Ratio ranking, we learnt that the Tot_qz and Tot_HWL attributes were the two most useful attributes for building a decision tree. It also revealed that those two attributes were good discriminants for the decision tree. Once the data attributes were selected, the J48 decision tree algorithm was executed using the 10-folds cross-validation method. The results of the experiment are discussed and presented in Section 5.2

Classification with Combined Dataset

In addition to the experiments using the Reflect dataset and students' assessment data, we also conducted classification experiments using the combination of Reflect data and students' assessment data as in Table 4.5. First, the usefulness of each data attribute in building a tree was examined by using Weka's Gain Ratio attribute evaluator along with the Ranker mechanism explained in the previous section. The best splitting criteria for each attribute were ranked accordingly as shown in Figure 4.5.

```
Attribute Evaluator (supervised, Class (nominal): 21 exam_cate):
        Gain Ratio feature evaluator

Ranked attributes:
 0.344      5 Tot_qz
 0.3147     3 Tot_HWL
 0.2306    19 Rflt_evd
 0.2077    20 Tot_selfAssess
 0.1834     7 n_task
 0.1823     8 n_lo
 0.1558     1 Tot_Lec_qz
 0.1347    12 L2_mem_diagram
 0.1274     9 HWLwk4
 0.1169     2 Lec_qz_attd
 0.1088    14 L8_scope
 0.1041     6 qz_attd
 0.0968     4 HWL_attd
 0.0948    11 HWLwk12
 0.0925    13 L5_pointers
 0.078     10 HWLwk8
 0         17 Example2
 0         18 Example3
 0         15 Example0
 0         16 Example1

Selected attributes: 5,3,19,20,7,8,1,12,9,2,14,6,4,11,13,10,17,18,15,16 : 20
```

Figure 4.5: Ranking of the combined data attributes in the Gain Ratio

Figure 4.5 shows that the Tot_qz attribute was ranked first with a splitting coefficient value 0.344. This is followed by the Tot_HWL attribute with splitting coefficient value 0.314. This indicates that these two variables were the main discriminators in forming branches of the tree. Meanwhile, four attributes namely Example2, Example3, Example0, and Example1 were ranked last in the list with the splitting coefficient values 0. This indicates that these four attributes were not important in building the decision tree. For this reason, when we run the experiments, they were omitted from the list of attributes. As a result, only 17 attributes were selected for the classification experiments. These attributes are listed in Table 4.10. After the attributes were selected, the classification experiments were executed with this dataset. The Weka's

J48 algorithm was executed with 10-folds cross-validation option selected. The result and discussion of the experiments will be presented in Section 5.3.

Table 4.10: List of all attributes used for building a model

Attribute Name	Description
L2_mem_diagram	Week 2 lecture quiz mark
L5_pointers	Week 5 lecture quiz mark
L8_scope	Week 8 lecture quiz mark
Tot_Lec_qz	Total lecture quiz marks (%)
Lec_qz_attd	Total lecture quiz attendances (%)
HWLwk4	Homework/Lab (prac exam) week 4
HWLwk8	Homework/Lab (prac exam) week 8
HWLwk12	Homework/Lab (prac exam) week 12
Tot_HWL	Total homework/lab marks (%)
HWL_attd	Total homework/lab attendances (%)
Tot_qz	Total quiz marks (%)
qz_attd	Total quiz attendances (%)
RFLT_evd	Evidence generated from student submission
Tot_selfAssess	Number of self assessments done in Reflect
n_task	Number of tasks done in the Reflect
n_lo	Number of learning objectives done in the Reflect
exam_cate	The categorical final exam mark

Association Rules Mining

The goal of an association rule mining algorithm is typically to identify association of items that occur together or sequences of items that occur frequently [32]. These associations and values are often represented by *IF-THEN* relationships.

In the present study, we applied the Apriori algorithm to discover which data attributes are related to each other. The objective was to find the relationships between the attributes that can be associated with positive or negative learning behaviour. The Apriori algorithm requires all data attributes to be nominal. Therefore, the numerical Reflect data attributes (described in Table 4.9) were converted into categorical form and renamed. The process is also called the discretisation process. These discretised attributes, as well as the categorical exam attributes, were used in the association rules analysis. Except for the exam_cate (exam) attribute that was manually categorised as explained in Section 4.3.2, all the other data attributes were discretised using the *equal-width* method. Using this method, the data was divided into a fixed number of intervals of equal (almost equal) length. As a result, all the data attributes were categorised into three intervals, namely, high, medium and low. The Reflect categorical data that were converted and their intervals are shown in 4.11. In addition to the attributes in this table in which they were converted from the Reflect dataset, we also converted some data attributes into categorical data from students' assessment dataset as shown in Table 4.12. These data would be used for the association rules analysis with combined dataset. The result of the experiments are discussed in Chapter 5.

Attribute	Categorical Attribute	Interval	Range
Rflt_evd	Rflt_cate	high >=66; medium >=33 and <66; low <33	min =1 max =99
Tot_selfAssess	tSelfAssess	high >=150; medium >=75 and<150; low <75	min =29 max =224
n_task	num_task	high >=18; medium >=8 and <18; low <8	min =1 max =24
n_lo	num_lo	high >=18; medium >=8 and <18; low <8	min =3 max =24
exam	exam_cate	high >=75; medium >=50 and <75; low <50	min =10 max =93

Table 4.11: Discretised the Reflect data attributes

Attribute	Categorical Attribute	Interval	Range (%)
Tot_qz	total_qz	high >=33; medium >=33 and <66; low <33	min =11 max =100
Tot_HWL	total_HWLab	high >=33; medium >=33 and <66; low <33	min =16 max =100

Table 4.12: Discretised the students' assessment data attributes

4.5 Conclusion

This chapter presented the core work of this study. It provided an overview of the approach and methodology by which the present study was conducted. We began by restating the research question and the subsidiary research question to be addressed in this study. The research question and the subsidiary research question were translated into six hypotheses that formed the thesis direction. Next, we introduced the subjects and data, followed by a discussion of how the data were pre-processed. Furthermore, we described the datasets and the application of EDM methods to carry out the experiments. As explained, before conducting a series of experiments using the Reflect dataset, we took several steps to process the student data. First, we needed to filter only the information about the student activities that interested us, such as the number of tasks, the number of learning objectives, the amount of evidence submitted in the Reflect system and the exam marks. Next, we needed to create a summary table that integrated the information at the student level. In addition, we needed to discretise all the numerical values in order to increase interpretation and comprehensibility. Finally, we described how the EDM methods including the K-means, J48 decision tree and Apriori algorithms were applied to the present study. It is expected that by deploying these data mining algorithms, we would be able to prove or disprove the hypotheses that would ultimately lead to the answers to the research questions as discussed next in Chapter 5.

Chapter 5

Results and Discussion

This chapter presents the results and discusses the findings of the present study. The results presented in this chapter were obtained from the exploration of the students' activity data (the Reflect data) and students' assessment data using the EDM methods discussed in the previous chapter. The presentation of the results and discussions in this chapter is arranged to address the hypotheses and how they were proved. There are two main parts of discussions in this chapter. The first part presents the results obtained from the exploration of the Reflect data followed by the discussion of the results. The second part discusses the results of the exploration of the students' assessment data and it is followed by the discussion of the results.

5.1 Exploration of the Reflect Dataset

5.1.1 Statistical Analysis

We performed a series of analyses by using both statistical methods and data mining techniques. First, we explored the dataset by using the statistical tool analysis from SPSS software[1]. SPSS is a well-known software package that offers a large number of methods and tools for statistical analyses. We utilised this software for the analysis of the normal distribution of the data and the correlation (the *Pearson correlation*) between the attributes in the dataset.

The Pearson correlation analysis, however, requires the data attribute to satisfy the *normality* and *linearity assumptions*. The normality assumption is examined by

[1] http://www.spss.com

projecting the variables into a histogram and observing whether the histogram formed a *bell curve graph*. The linearity assumption of two related variables is verified by checking a scatter plot generated from those two variables. Furthermore, a scatter plot is used to investigate whether the variables have *homoscedasticity*. This can be known by reviewing whether the dots are evenly distributed in the graph. We examined the variables and found that they satisfied the normality and linearity assumptions, as discussed in the following section.

First Hypothesis

The primary focus of the current project is to examine how students have used Reflect to enhance their knowledge in order to better prepare for the exam. Therefore, most of the statistical and data mining analyses in the thesis are centred around the exam marks. It is important to do exploratory analysis of the data in order to gain initial insights and knowledge about the data. Figure 5.1 shows a histogram of a normal distribution curve related to the exam mark. As we can see from the graph, most of the students' exam marks were within the mean range (53.15). The distribution of the mean of the students' exam marks formed a bell-shaped curve that is known as the normal distribution or Gaussian function. By observing the graph, we can also understand that only a small number of students were able to achieve an exam mark equal or higher than 80. However, since the exam mark is not the only component to determine the students' final grade, students who have average exam marks can still achieve a good final grade at the end of semester provided that they perform better in other assessment components.

The normality assumption was also observed for the RFLT evidence (Rflt_evd) variable that was introduced in Chapter 3. Figure 5.2 shows that the Rflt_evd attribute value is normally distributed and satisfied the normality assumption.

After all the required variables were checked to satisfy the normality assumption, the next step was to investigate whether there were relationships between those variables and whether the relationships were linear. The most common way to do this is by plotting the variables into a scatter plot graph.

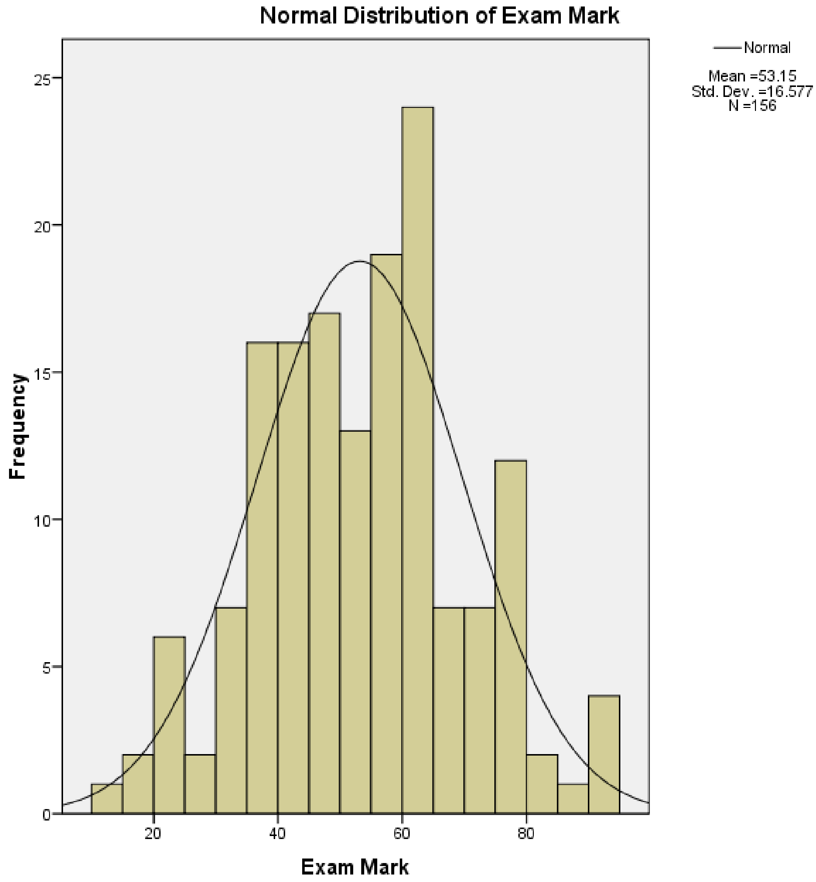


Figure 5.1: Normal distribution graph of exam

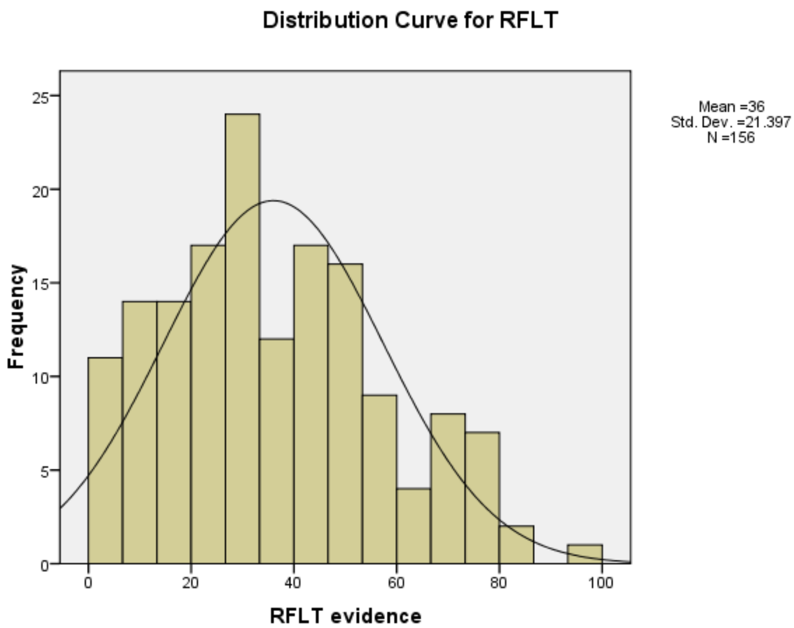


Figure 5.2: Normal distribution graph of Rflt_evd

For example, to observe if there was a linear relationship between the Rflt_evd and the exam mark, a scatter plot of those two variables was created (Figure 5.3). The graph revealed that there was a strong positive relationship between the Rflt_evd and the exam mark. From the cumulative Rflt_evd graph as shown in Figure 5.4, it is also revealed that around 80 percent of the students submitted 50 or less RFLT evidence to the system. The average (mean) of the RFLT evidence submitted by the students was 33, while the minimum and the maximum number of RFLT evidence submission were 1 and 99, respectively.

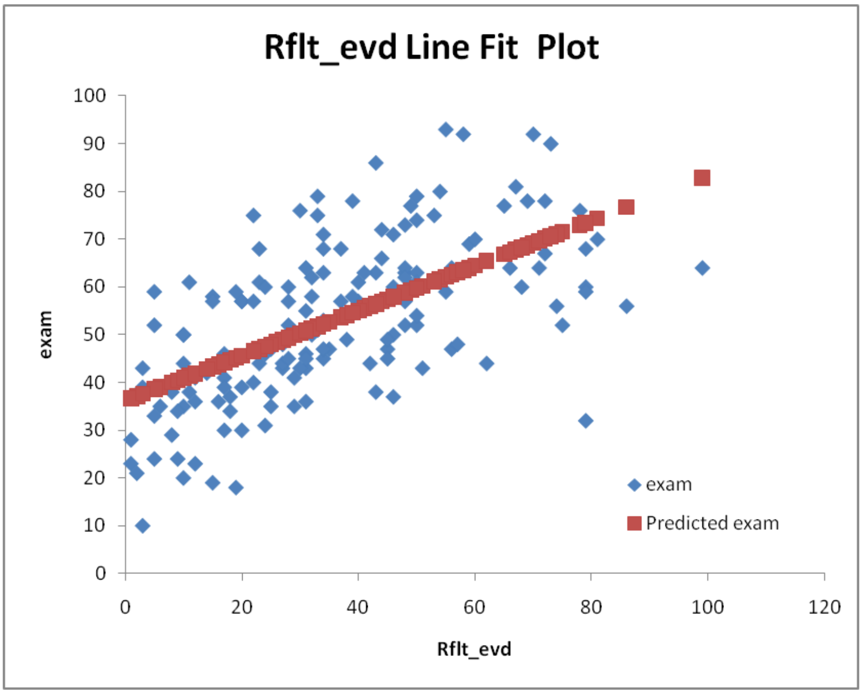


Figure 5.3: Scatter plot of Rflt_evd

Furthermore, to determine the strength of the relationships between the Rflt_evd and the exam mark, a correlation analysis of the two variables was conducted. This analysis was performed by using the *Pearson correlation coefficient* analysis tool available in SPSS.

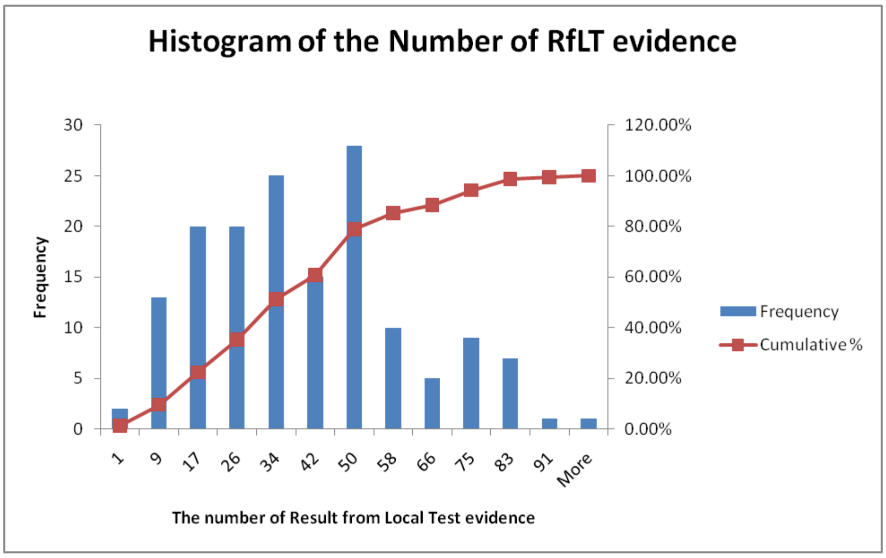


Figure 5.4: Cumulative percentages of Rflt_evd

The result shows that the relationships between the two attributes were strongly positive as shown by the Pearson correlation result in Table 5.1. The Pearson correlation coefficient relationship between the Rflt_evd and the exam marks was reasonably high ($r=0.618$; $p<0.01$). In a general interpretation, this means that the more RFLT evidence submitted by a student, the more likely the student will achieve a better exam mark.

Table 5.1: Correlation between the Rflt_evd and the exam

		Rflt_evd	exam
Rflt_ evd	Pearson Correlation	1	.608**
	Sig. (2-tailed)		.000
	N	156	156

** Correlation is significant at the 0.01 level (2-tailed).

The results can be used as evidence that submitting more RFLT evidence in Reflect increases the chance of performing well in the exam. One possible reason for this relation may be because submitting a correct code for a task requires students to understand the task and the learning objectives associated with that task. This is understandable if we remember that the Reflect system would only record the RFLT evidence if the programming code submitted by the students passed certain criteria set up by the teacher or the tutor.

The results of the statistical analysis by graphs and the Pearson correlation analysis suggested a strong positive correlation between the RFLT evidence submission in the Reflect system and the exam mark. The figures indicate very encouraging support for the *first hypothesis*, namely, that "*Students who submitted RFLT evidence above the class average in the Reflect system are more likely to perform well in the exam*".

Second Hypothesis

The second hypothesis we examined is "*The total number of the self-assessments performed in the Reflect system correlates positively to the exam performance*".

In this observation, we were interested to learn whether there was any relationship between the variable of total self-assessment (Tot_selfAssess) and exam (the exam mark). First, the Tot_selfAssess variable was checked regarding whether it satisfied the normality assumption. This was done by plotting the variable into a normal distribution graph. The result shows that the Tot_selfAssess variable satisfied the normality assumption. The fit-line of the normally distributed value is clearly seen from the graph (Figure 5.5).

Second, a scatter plot was used to survey a distribution of the Tot_selfAssess variable. Figure 5.6 shows that the instances (represented by blue dots) of the Tot_selfAssess variable were evenly distributed and centred around a linear fit-line. This means that the variable Tot_selfAssess satisfed the assumption of homogeneity. Figure 5.6 also shows that there is linear correlation between the Tot_selfAssess and exam. This means that the increase in the number of self-assessments submitted to the Reflect system would have an impact on the exam mark, that is the exam mark would be increased.

Finally, the correlation between Tot_selfAssess variable and the exam mark was examined by using the Pearson correlation coefficient as shown in Table 5.2. The result of the Pearson correlation analysis suggested that there was a positive correlation between the Total number of self assessments performed in the Reflect system and the exam mark as shown by the correlation coefficient between the two variables ($r=0.316$; $p<0.01$).

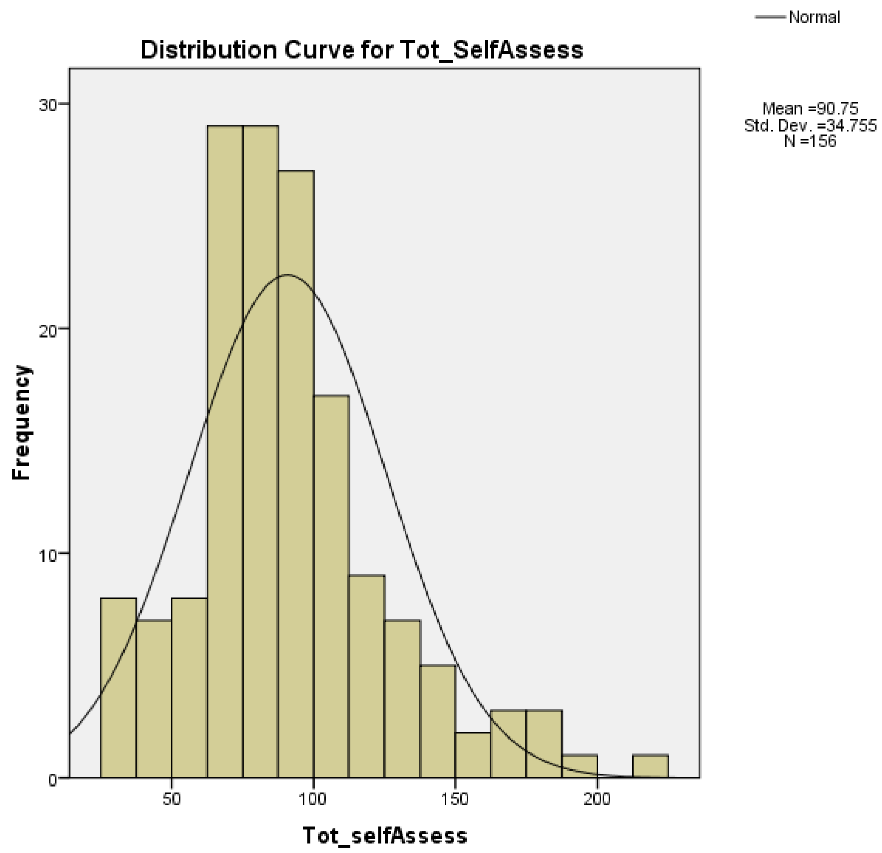


Figure 5.5: Normal distribution graph of Tot_selfAssess

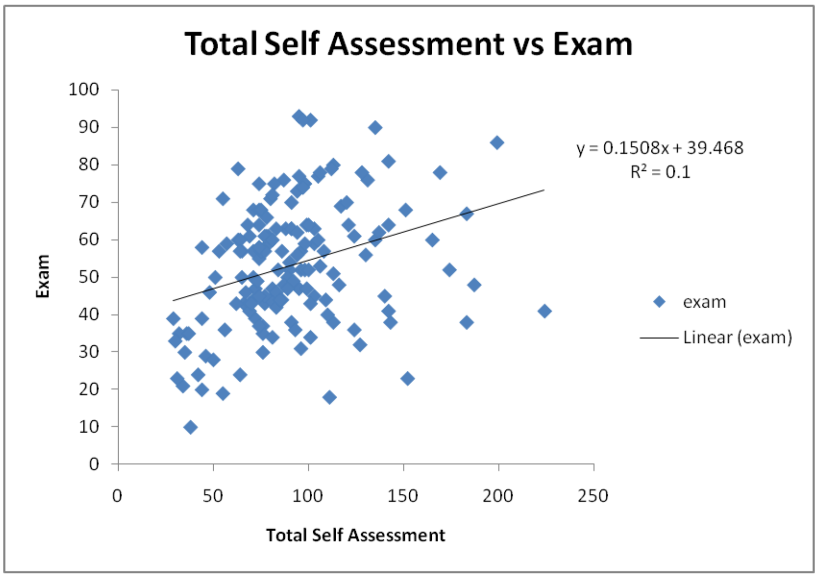


Figure 5.6: Scatter plot of Tot_SelfAssess and the exam

Table 5.2: Correlation between total self-assessment and exam mark

		exam	Tot_selfAssess
exam	Pearson Correlation	1	.316**
	Sig. (2-tailed)		.000
	N	156	156

** Correlation is significant at the 0.01 level (2-tailed).

The scatter plot of the total self-assessment submissions (Tot_selfAssess) and the exam (exam mark) also revealed that the highest exam mark was reached when the number of submissions for self-assessment was around 100. Beyond that point, the exam marks decreased. This means that the number of self-assessments submitted in Reflect did not always correlate positively to an increase of the exam mark. It seems that there is a critical point (in this case 100 submissions) for the exam mark to increase in relation to the number of self-assessments submitted in Reflect. After this point, the exam mark variable will not increase further regardless of how many more self-assessments are submitted. A possible reason for this is that the students who made an excessive number of submissions (more than 100) may not have been thinking about the problem carefully and may have been guessing the answers, while students who made fewer submissions could be either advanced students who were able to solve problems quickly or poor students who gave up quickly. This observation is in line with the result of a study by Allevato et al. [2]. That study was about a group of students who used a learning system intensively. It found that the students who made more submissions to the system compared to those who made less did not necessarily achieve a higher score in the final exam. Their study may provide insights to explain how the students can submit a lot of work in Reflect yet perform poorly in the exam.

The results of the statistical analysis by graphs and the Pearson correlation analysis suggested a positive correlation between the total self-assessment submission in the Reflect system and the exam mark. The results suggest an encouraging support for the second hypothesis, namely, that "*The total number of the self-assessments performed in the Reflect system correlates positively to the exam performance*".

5.1.2 Clustering Analysis

Third Hypothesis

The third hypothesis was developed to examine whether there were any relationships between the independent practice of the example solutions and the submission of the RFLT evidence and the exam mark. The students were encouraged to practise their programming skills regularly in the Reflect system: for example by accessing the example solutions and submitting their own code for evaluation. It was expected that the students should be practising as much as possible in the Reflect system. By practising the exercises in the Reflect system, students would develop the knowledge and skills necessary to perform more complex tasks in the student submission category (RFLT evidence). We hypothesised that if students tried fewer example solutions, it would affect their knowledge and skills gains, and thus they would not be able to submit sufficient number of RFLT evidence submissions to the system. As a result, the students would be less prepared for the exam. Therefore, the third hypothesis was formed as: *"Fewer submission of example solutions and RFLT evidence in the Reflect system would lead to poor exam performance"* ,

For this hypothesis, we selected five numerical attributes related to students' submission of example solutions and students' submission of RFLT evidence. The attributes selected were Example0, Example1, Example2, Example3 and Rflt_evd. Once the attributes were selected, the Weka SimpleKMeans application was executed and the default Euclidean distance option was chosen. The experiments were run a number of times by varying the number of clusters (k) from 2 to 4. The results were compared and we found that the best result was obtained when the (k=3) setting was chosen for the number of clusters.

At this stage, we also wanted to compare the result of the clustering in which the data attributes were normalised with the result of clustering without attribute normalisation. Therefore, in the first run the attributes were not normalised, while in the second run, the attributes were normalised. In the end, it was found that the results were not much different in terms of the distribution of contents and members for each cluster. For example, we found that the number of instances for each cluster where the attributes were normalised were similar to the results of the clusters where the attributes were not normalised. The small size of the data may be responsible for these results. In other words, the larger dataset may produce less similarity in the results.

One of the outputs of the clustering experiments with the normalised attributes is shown in Figure 5.7. The figure shows three clusters that grouped students according to the example solutions and RFLT evidence they submitted. The main distinctive feature from the figure is that the RFLT evidence correlated positively to the example solutions 0 to 2. If the example solutions on average were high, the RFLT evidence was also high as in cluster 1.

```
kMeans
======

Number of iterations: 5
Within cluster sum of squared errors: 14.433659941509887
Missing values globally replaced with mean/mode

Cluster centroids:
                           Cluster#
Attribute      Full Data          0          1          2
                   (156)       (38)       (47)       (71)
=========================================================
Example0          0.2711     0.3096     0.5054     0.0955
Example1           0.333      0.406     0.5459     0.1531
Example2           0.315     0.3898     0.4603     0.1788
Example3          0.0874     0.1542     0.0922     0.0486
Rflt_evd          0.3571     0.2787     0.5979     0.2397



Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       38 ( 24%)
1       47 ( 30%)
2       71 ( 46%)
```

Figure 5.7: Clustering analysis related to students' submissions .

On the other hand, when the example solutions were relatively small, the RFLT evidence was also small as in cluster 2. For this reason, it is logical to associate the example solution with the RFLT evidence, that is they were related to each other and they association had an impact on the exam performance. From the clustering results shown in Figure 5.7, the students can be grouped into 3 categories as summarised in Table 5.5 .

Beside the clustering analysis, the analysis of variance (ANOVA) is also conducted to investigate whether there are statistically significant different of the means of the exam mark among the clusters. The null hypothesis of this test stated that there are no real different in the means of the exam mark between the clusters. But we would like to proof that there are significant different in the exam means among the clusters. First, the summary of cluster variance is shown in Table 5.3. Then the result of ANOVA test is presented in Table 5.4. The result of ANOVA test suggested that there are significant different among the means of exam mark within the three clusters. This can be assessed by comparing the reported p-value from the ANOVA test with the chosen alpha level (.05). Here, the conclusion can be withdrawn that the reported p-value from the test (.000) is much smaller than the chosen alpha level (.05). Therefore, we can reject the null hypothesis in favour of the the hypothesis that there are statistically significant different in the means of exam mark among the clusters. This result suggested that in general there are positive correlations between submitting the example solutions and the exam mark.

Table 5.3: Summary of cluster variance

Groups	Count	Sum	Average	Variance
Cluster0	38	1888	49.684	215.844
Cluster1	47	3079	65.511	189.777
Cluster2	71	3325	46.831	220.057

Table 5.4: The ANOVA test

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	10472.38	2	5236.19	24.942	.000	3.0552
Within Groups	32119.93	153	209.93			
Total	42592.31	155				

Useful Information related to Learning Behaviour

The clustering analysis result suggested that the students who performed fewer example solutions in the Reflect system were also submitted fewer RLFT evidence (the members of cluster 2 that constituted 46% of the population). These behaviours can be identified as disengagement in learning.

Table 5.5: Groups of students distinguished by example solutions

Cluster size and (percentages)	Example Solutions	Rflt Evidence	Group Label
47 students (30%)	Average high number of submissions	High number of submissions	Regular user
71 students (46%)	Average medium number of submissions	Medium number of submissions	Moderate user
38 students (24%)	Average low number of submissions	Low number of submissions	Casual user

On the other hand, the more regular Reflect users who perform on average high number of example solution submissions were also submitted high number of RFLT evidence (the members of cluster 1 that constituted 30% of the population). These behaviours can be identified as positive engagement in learning that can promote the reflection for learning.

Table 5.5, shows that the majority of the students were moderate Reflect users (comprised of 46% (71 students) of the total students). This means that most of the students in the class performed on average a medium number of submissions in both example solutions and RFLT evidence.

From a pedagogical perspective, it can be suggested that more attention should be paid to the group of "casual user" as they made on average low submission in both example solutions and RFLT evidence as this could be identify as a group of students who less engage in learning. . The low submissions in both categories suggest that the students might not learn sufficient programming skills and knowledge from the course. This could lead to poor performance in the exam.

The result of this clustering analysis supports the third hypothesis that *"The fewer submission of example solutions and RFLT evidence in the Reflect system would lead* *to poor exam performance"*

5.1.3 Classification Analysis

Beside the statistical and clustering analyses, this research also used classification analysis to classify the students. The objective was to build a model (or classifier) that can classify students according to their exam performance. The model can be used to classify new students and predict their exam performance.

The dataset contained eight data attributes. We utilised Gain Ratio attributes evaluation along with a search mechanism called Ranker to evaluate and the attributes that would be used to first split the decision tree. This process and chosen attributes was previously discussed in Section 4.4.2. According to the Gain Ratio list Rflt_evd and Tot_selfAssess were the top two most important attributes for building a decision tree.

For this experiment, the J48 decision tree algorithm was used to perform clustering analysis as explained in Section 4.4.2. Since the J48 algorithm requires that the class attribute must be nominal, a numerical dataset with a nominal class was used. The nominal class attribute is the exam marks that have been grouped into three ordinal categories, namely, high (if the mark >75), moderate (if the mark was between >=50 and <75), and low (if the mark <55). The reason for this categorisation and the chosen interval was discussed in Section 4.4.2.

After the attributes were selected, a series of classification experiments were conducted using Weka's J48 algorithm. The 10-folds cross-validation method was chosen to separate the data used for training sets and test sets as explained in Section 4.4.2. The Weka's J48 classifier enable users to set-up the minimum number of instances (minNumObj) per leaf to be considered as the splitting attribute. We found that the percentage of correctly classified instances (accuracy) of a model increased when the minNumObj threshold were increased. However, the increases were not continue after the minNumObj reached 12. At this point, the best model was obtained as shown in Figure 5.8.

From the classification experiment as shown in Figure 5.8, the accuracy of the model was 61.53%. From the result, the accuracy of the model was promising if it compared to the baseline accuracy. There are many baseline accuracy measures that can be used for a comparison, but the one that is simple and easy to use is the prediction of the majority class. The prediction of the majority class for the dataset is 44.8%. This is the prediction of 70 students to be in the "moderate" class out of 156 instances. The dataset contains some missing values that were intentionally left as it is; for example in Example0, Example1, Example2 and Example3 data attributes. The accuracy of the model can be increased by performing pre-processing again to remove the missing values in the dataset. However, in some educational data mining research and in the present study, this is not done in order to learn potentially important information that may exist in the missing data.

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

Rflt_evd <= 31
|   Tot_selfAssess <= 97
|   |   n_lo <= 12: Low (39.0/8.0)
|   |   n_lo > 12: Moderate (20.0/8.0)
|   Tot_selfAssess > 97: Low (14.0)
Rflt_evd > 31: Moderate (83.0/32.0)

Number of Leaves  :     4

Size of the tree :      7


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          96               61.5385 %
Incorrectly Classified Instances        60               38.4615 %
Kappa statistic                          0.3114
Mean absolute error                      0.3255
Root mean squared error                  0.422
Relative absolute error                 81.2675 %
Root relative squared error             94.3826 %
Total Number of Instances              156
```

Figure 5.8: The J48 tree analysis

The J48 decision tree algorithm was able to predict all the class attributes except for the "High" class that was not shown in the tree. In this case, the J48 algorithm may consider that the High class is not an important determinant for the decision tree thus the High class prediction is not shown in the tree. Since not all the rules extracted from a decision tree are interesting, we could assume that the particular rules related to the High class attribute are not particularly interesting in this case. Hence, we more determine to extract the rules more interesting from the two other classes (Moderate and Low class).

Beside the statistical figures, the Weka's J48 can also visualise the result of a decision tree as a tree model as shown in Figure 5.9. This visualised tree helps to understand the classification results. From this tree model, we can see that the size of the tree is seven and the leaves is four.

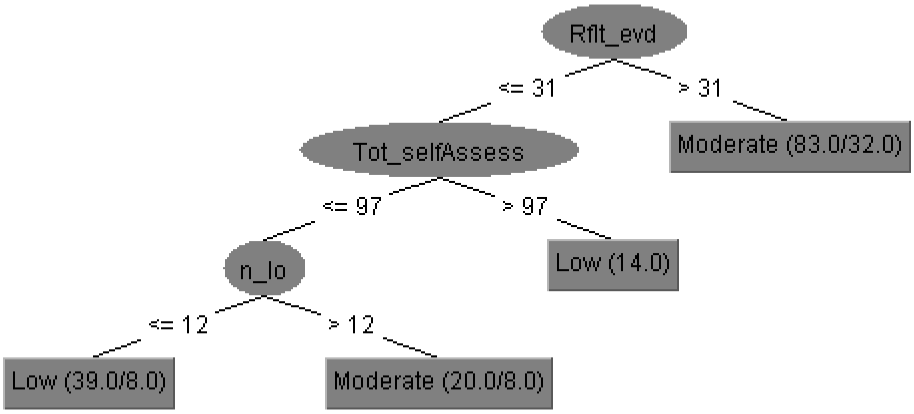


Figure 5.9: Visualisation of the tree model

The tree in Figure 5.9 can be easily converted to rules that are made up of the *antecedent* and the *consequent*. The antecedent of the rules consists of conditions that must be satisfied by the node on the path from the root to the leaf. The consequent of the rules consists of the class assigned by the leaf. One rule is derived from one leaf. For each rule, the antecedent and consequent form an *IF-THEN* association. However, not all the rules were found to be interesting. Therefore, not all the rules from the decision tree can be used. Some of the interesting rules extracted from the decision tree in Figure 5.9 are summarised as follows:

Rule 1: IF (Rflt_evd <= 31) AND (Tot_selfAssess <= 97) AND (n_lo <= 12) THEN exam = Low

Rule 2: IF (Rflt_evd <= 31) AND (Tot_selfAssess <= 97) AND (n_lo > 12) THEN exam = Moderate

Rule 4: IF (Rflt_evd > 31) THEN exam = Moderate

There is one rule that was uninteresting and confusing:

Rule 3: IF (Rflt_evd <= 31) AND (Tot_selfAssess > 97) THEN exam = Low

Useful Information related to Learning Behaviour

The interesting rules generated by the decision tree contain valuable information for the teacher or course coordinator. For example, from the rules, it was found that the RFLT evidence submitted and the total self-assessments performed in the Reflect system played an important role in the students' success in the exam. It was found that the students who submitted their RFLT evidence at least 32 times were more likely to achieve a "moderate" or higher mark in the exam (rule 4). This behaviour can be identified as a positive learning behaviour that promote reflective learning. The other students who made fewer RFLT submissions (below 32) were likely to obtain a "low" or "moderate" exam mark depending on the other submissions such as total self-assessments and the number of learning objectives performed in the Reflect system (rules 1 and 2). These behaviours can be identified as negative learning behaviours that can affect learning.

This is some pedagogical information that can be extracted from the decision tree algorithm and can be very useful for the teacher and course coordinator. By understanding this information early in the semester, the course coordinator can take precautionary measures for the group of students who fell in this category. For example, providing extra tutorials for them or encouraging them to use the Reflect system more could promote more learning by reflection.

The result of classification analysis presented in this section suggested that a strong positive correlation between the RFLT evidence (Rflt_evd) and the total self-assessement (Tot_selfAssess) submissions in the Reflect system and the exam mark. The figures indicate very encouraging support for *the first hypothesis* and *the second hypothesis* that have been discussed in detail in Chapter 4 and at the beginning of this Section.

5.1.4 Association Rules Analysis

The Apriori algorithm was used to find rules that associate students's learning behaviours in using the Reflect system. Since the algorithm takes only nominal or ordinal attributes, the discretisation method had to be employed to convert the numerical Reflect dataset into nominal values. At first, we used the *discretise* function from Weka to categorise the numeric value of the dataset but we found that the method did not satisfy our need to divide each numeric attribute into three intervals of equal length (high, medium, and low). Therefore, the equal-width method was used to divide the data into intervals with equal or almost equal lengths (high, medium and low). The detailed information about the Reflect data attributes that were transformed into categorical form was discussed in Section 4.4.2. In this experiment, four categorical attributes from Table 4.11, namely Rflt_cate, tSelfAssess, num_task, and num_lo form Table and the exam_cate were used.

After the nominal dataset were ready, the Apriori algorithm was executed using this dataset. In the first run, the default values of 0.1 (10%) for minimum support and 0.9 (90%) for confidence were chosen. As a result, ten rules were found as shown in Figure 5.10. In the second run, the minimum support was increased to 0.2 (20%) and the confidence was increased to 0.9 (90%). As consequence, the number of rules decreased to only four rules as shown in Figure 5.11.

Some studies reported in the literature have suggested that some strong rules could be misleading and affect the interpretation [32]. In certain cases, the support and confidence alone are insufficient to determine the interestingness of rules.

```
Best rules found:

 1. num_task=High 38 ==> num_lo=High 38    conf:(1)
 2. num_task=Low exam_cate=Low 33 ==> Rflt-cate=Low 33    conf:(1)
 3. num_task=High tSelfAssess=High 24 ==> num_lo=High 24    conf:(1)
 4. num_lo=Low 22 ==> num_task=Low 22    conf:(1)
 5. num_lo=Low 22 ==> Rflt-cate=Low 22    conf:(1)
 6. num_task=High exam_cate=Moderate 22 ==> num_lo=High 22    conf:(1)
 7. num_lo=Low Rflt-cate=Low 22 ==> num_task=Low 22    conf:(1)
 8. num_task=Low num_lo=Low 22 ==> Rflt-cate=Low 22    conf:(1)
 9. num_lo=Low 22 ==> num_task=Low Rflt-cate=Low 22    conf:(1)
10. num_task=High Rflt-cate=Medium 20 ==> num_lo=High 20    conf:(1)
```

Figure 5.10: Association rules with minimum support 0.1 and min confidence 0.9

```
Apriori
=======

Minimum support: 0.2 (31 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 16

Generated sets of large itemsets:

Size of set of large itemsets L(1): 11

Size of set of large itemsets L(2): 23

Size of set of large itemsets L(3): 5

Best rules found:

 1. num_task=High 38 ==> num_lo=High 38    conf:(1)
 2. num_task=Low exam_cate=Low 33 ==> Rflt-cate=Low 33    conf:(1)
 3. num_task=Low 42 ==> Rflt-cate=Low 41    conf:(0.98)
 4. num_task=Medium num_lo=Medium 41 ==> tSelfAssess=Medium 37    conf:(0.9)
```

Figure 5.11: Association rules with minimum support 0.2 and minimum confident 0.9

Therefore, another measurement of interestingness such as a correlation analysis called *lift* should be used to accompany the support and confidence. Lift is a correlation measurement between two or more item-sets (attributes); for example, item-sets A and B in which the occurrence of item-set A is independent of the occurrence of item-set B. Lift is measured by dividing the confidence by the support of that relationship. If the lift value is less than 1, then the occurrence of A is negatively related to the occurrence of B. If the lift value is greater than 1, then the item-sets A and B are positively correlated and if the lift value is equal to 1, then the item-sets A and B are independent, meaning there are no relationships between these two item-sets.

In the final run of association rule analysis, the lift correlation measurement was set to a minimum value of 1.1 in Weka and the rules that resulted from this experiment are shown in Figure 5.12.

```
Apriori
=======

Minimum support: 0.25 (39 instances)
Minimum metric <lift>: 1.1
Number of cycles performed: 15

Generated sets of large itemsets:

Size of set of large itemsets L(1): 10

Size of set of large itemsets L(2): 13

Best rules found:

 1. num_task=Low 42 ==> Rflt-cate=Low 41    conf:(0.98) < lift:(2)> lev:(0.1
 2. Rflt-cate=Low 76 ==> num_task=Low 41    conf:(0.54) < lift:(2)> lev:(0.1
 3. num_lo=High 73 ==> Rflt-cate=Medium 46    conf:(0.63) < lift:(1.61)> lev:
 4. Rflt-cate=Medium 61 ==> num_lo=High 46    conf:(0.75) < lift:(1.61)> lev:
 5. Rflt-cate=Low 76 ==> exam_cate=Low 52    conf:(0.68) < lift:(1.59)> lev:
 6. exam_cate=Low 67 ==> Rflt-cate=Low 52    conf:(0.78) < lift:(1.59)> lev:
 7. num_lo=Medium 61 ==> Rflt-cate=Low 46    conf:(0.75) < lift:(1.55)> lev:
 8. Rflt-cate=Low 76 ==> num_lo=Medium 46    conf:(0.61) < lift:(1.55)> lev:
 9. num_lo=Medium 61 ==> tSelfAssess=Medium 51    conf:(0.84) < lift:(1.39)>
10. tSelfAssess=Medium 94 ==> num_lo=Medium 51    conf:(0.54) < lift:(1.39)>
```

Figure 5.12: Association rules with lift correlation analysis

Figure 5.12 shows that the minimum support for the rules that was used by the algorithm is 0.25 (25%) and the minimum lift is 1.1. This means that all the rules resulted from this experiment were positively correlated as events (lift is greater than 1). Ten rules were found in this experiment but not all of them were interesting. As discussed in the previous section, besides the support and confidence, the lift measurement was also used to determine which rules were the most interesting. The result reveals that the most interesting rules were rule number 1 and rule number 9 (the confidence values were 0.98 and 0.84, respectively, and the lift values were 2 and 1.39, respectively).

Useful Information related to Learning Behaviour

The first rule suggests that the number of tasks (num_task) performed by the students in the Reflect system was positively correlated to the amount of RFLT evidence (Rflt-cate) submitted by the students to the the Reflect system. If the students did not practise the sufficient tasks in the Reflect system, it was more likely that they would not be able to find solutions of the programming tasks set-up by the teacher. As consequence, they would not be able to submit a correct solution that would be recorded as RFLT evidence in the Reflect system.

The ninth rule suggests that the number of learning objectives (num_lo) was correlated positively with the number of total self-assessments (tSelfAssess) performed by the students in the Reflect system. This suggests that students should look at and understand the learning objectives of a task before they can assess themselves. As discussed in Section 3.3, self-assessment requires students to understand the concepts in the learning objectives and self-assess their understanding of that concept in the Reflect system.

The fifth rule is also interesting because it revealed a direct correlation between submitting RFLT evidence and the exam mark. It suggested that students who were not able to submit sufficient RFLT evidence were more likely to achieve a fail or low exam mark. This rule is related directly and strongly supports the *first hypothesis* that was discussed in the beginning of this section. The rule number one and the rule number nine also support the *first hypothesis* and *the second hypothesis* that emphasised the impact of RFLT evidence and total self-assessment on the exam mark.

5.2 Exploration of Student Assessment Dataset

EDM methods was also used to investigate the impact of lecture quizzes and lab sessions on student performance as reflected in the final exam mark. This analysis used the students' assessment dataset from the course coordinator as described in Section 4.2.2. This additional study was conducted in order to find the relationships between the class activities given by the lecturer and student performance. It sought the answers to the following two questions:

1. What is the impact of total quiz marks to exam performance?

The fourth hypothesis introduced at the beginning of Chapter 4 was developed to answer this question. Therefore, the fourth hypothesis was formed as: *"Students who attend and achieve good marks on all quizzes have a better chance at performing well in final exam"*

This hypothesis was developed to investigate the relationships between the students' quiz marks and their exam marks. We hypothesised that there was a strong positive correlation between students who did not miss a class and worked hard to achieve good marks on all quizzes and performing well in the final exam.

2. *What is the impact of weekly homework/lab sessions on exam performance?*

The fifth hypothesis was developed to answer this question. Therefore, the fifth hypothesis was formed as: "*Students who miss weekly homework/lab exercises are at an increased of risk of performing poorly in the final exam*".

The hypothesis was developed to evaluate the relationships between students' failure to attend the weekly homework/practical lab sessions (that were aggregated into the total homework/lab marks) and their performance in the final exam.

5.2.1 Statistical Analysis

A descriptive analysis was run to gain a perspective about the data attributes that would be used for data mining experiments. Selected data attributes that were associated with student assessments and attendances were chosen. These included the students' marks and attendances for quizzes and lecture quizzes, homework/lab sessions and final exam marks. Some redundant data were not included in the list, for example, the total lecture quizzes (Tot_Lec_qz) and HWLwk4, HWLwk8, HWLwk12 as explained in Section 4.4.2 . A complete statistical descriptive analysis is shown in Table 5.6. The classification analysis were performed using selected attributes from this dataset except for the exam mark attribute that was categorised into tree intervals of "high", "moderate" and "low".

Table 5.6: Descriptive statistics of the attributes

Attribute	N	Minimum	Maximum	Mean	Std. Dev
L2_mem_diagram	95	1	5	3.61	1.07
L5_pointer	109	0	5	3.35	1.26
L8_scope	77	1	7	3.77	1.34
Lec_qz_attd	156	0%	100%	60.04%	35.40%
Tot_HWL	156	16%	100%	70.67%	21.49%
HWL_attd	156	20%	100%	84.29%	20.07%
Tot_qz	156	11%	100%	68.4%	23.4%
qz_attd	156	20%	100%	84.3%	19.3%
exam	156	10	93	53.15	16.58

5.2.2 Classification Analysis

The Weka's J48 algorithm was also used for classification analysis of students' assessment data. The superiority of the J48 algorithm was discussed in Section 2.6.2 and how this algorithm was applied to the data was explained in Section 4.4.2 in relation to the research approach. This section discusses the results of the application of the decision tree algorithm to the students' assessment data.

For this analysis, a numerical dataset with a categorical class label was used for this classification analyses. The class label was the exam mark which was categorised into ordinal values as explained in Section 4.3.2. The description of list of attributes for the students' assessments data was presented in Table 4.3. For the experiments, nine data attributes including exam were selected. We used Gain Ratio attributes evaluation along with a search mechanism called Ranker to select the most useful attribute. In Section 4.4.2 we have discussed the process and chosen attributes using Gain Ratio attributes evaluator. According to the Gain Ratio list total quiz marks (Tot_qz) and total homework/lab marks (Tot_HWL) were the top two most important attributes for building a decision tree.

Next, the classification experiments were executed a number of times. The first experiment obtained 69.2% of the accuracy. This meant that out of 156 instances, only 108 instances were correctly classified into the classes. The other 48 instances were incorrectly labelled into the other classes.

In second experiment, the minimum number of instances per leaf (minNumObj) was increased from 2 to 4 but the percentage of accuracy was steady. The experiment was repeated with the number of instances per leaf was increased from 4 to 8. As a result, we found that the best accuracy for the model is 72.43% with the minNumObj set to 6 as shown in Figure 5.13.

From the experiment result shown in Figure 5.13, we learn that the correctly classified instances (accuracy) of the model is 72.43%. The accuracy of the model was high if it compared to the baseline accuracy. The simple baseline accuracy is the prediction of the majority class in which the accuracy is 44.8%. This means that the model built by the algorithm is a good classifier that can be used to predict new students' exam marks.

From the decision tree, a number of interesting rules were extracted as follows:

Rule 1: IF (Tot_qz <= 0.74) THEN exam = Low

Rule 2: IF (Tot_qz > 0.74) AND (Tot_qz <= 0.96) THEN exam = Moderate

Rule 3: IF (Tot_qz > 0.96) AND (Tot_HWL <= 0.94) THEN exam = Moderate

Rule 4: IF (Tot_qz > 0.96) AND (Tot_HWL > 0.94) THEN exam = High

Useful Information related to Learning Behaviour

The results suggested that the total quiz mark (Tot_qz) was the main discriminator of the final marks, followed by the the total homework/labs marks. The rules generated by the J48 algorithm revealed the characteristics of each group of students. For example, a student should achieve at least 74% of the total quiz marks to reduce the risk of failure in the final exam (rule 1). Meanwhile, a student who achieved 96% of the total quiz marks and at least 94% of the total homework/labs marks was directly classified as "High" for the exam (rule 4), meaning he/she will be among the students who are likely to achieve a good mark in their final exam. The other groups of students were classified "Moderate" based on the total marks of the quizzes (Tot_qz) and the homework/lab sessions (Tot_HWL) (rule 2 and 3).

Some pedagogical information can be retrieved from the results, for example the critical separation points were set for each class; namely, 74% or 96% of total quizzes and 80% of the total homework/labs (Tot_HWL). These separation points were set automatically by the J48 classifier.

This information would be very useful for a course coordinator; for example, the classifier can be used to predict the final exam marks of new students based on their quiz marks or to encourage students to be more engage in the types of activities similar to lab exercises.

The result of this classification analysis using the students' assessment data suggested a strong positive correlation between the total quiz marks and the total homework/labs marks and the exam mark. This result supports *the fourth* and *the fifth hypothesis.*

```
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

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

Tot_qz <= 0.74: Low (85.0/25.0)
Tot_qz > 0.74
|   Tot_qz <= 0.96: Moderate (52.0/13.0)
|   Tot_qz > 0.96
|   |   Tot_HWL <= 0.94: Moderate (9.0/3.0)
|   |   Tot_HWL > 0.94: High (10.0)

Number of Leaves  :     4

Size of the tree :      7


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         113               72.4359 %
Incorrectly Classified Instances        43               27.5641 %
Kappa statistic                          0.5281
Mean absolute error                      0.2603
Root mean squared error                  0.3691
Relative absolute error                 64.9724 %
Root relative squared error             82.566  %
Total Number of Instances              156
```

Figure 5.13: Decision tree built from the students' assessment data

5.3 Exploration of Combined Dataset

5.3.1 Classification Analysis

After the experiments with the students' activity data (the Reflect dataset) and the students' self-assessmeent data, we also conducted classification experiments with the combined dataset from the students' activity data and the student's assessment data. The objective is to examine the main attributes in this dataset that contribute to the students' exam performance. Initially, the dataset contained 21 attributes. However, not all the attributes were equally important or useful in predicting the target. Therefore, the Gain Ratio attribute evaluation along with a search mechanism called Ranker was deployed to rank each attribute according to its usefulness in building a model. The list of ranked attributes is shown in Figure 4.5. According to the list of ranked attributes, Tot_qz, Tot_HWL, Rflt_evd and Tot_selfAssess were the top four most important attributes for building a decision tree. Example0, Example1, Example2 and Example3 were located at the bottom of the list, with each of them having a zero value for the coefficient ranking. Based on this, these four attributes were removed from the dataset. At this stage, the dataset contained 17 attributes that could be used for classification analysis. Once the dataset was ready, a number of classification analyses were performed in Weka for which the 10-fold cross-validation method was chosen as explained in Section 4.4.2.

The experiments began by executing the J48 algorithm with the dataset containing 17 attributes. The result was a decision tree with 14 leaves and a tree size of 22. This tree shows correctly classified instances of 69.87%. This percentage of correctly classified instances (accuracy) was quite promising if it compared to the baseline accuracy which is the prediction of the majority class. The prediction of the majority class for the dataset was 44.8% of the 156 instances.

To increase the classifier accuracy usually measured by the percentage of correctly classified instances, a series of experiments was conducted by reducing the number of attributes one by one starting from the attribute in the last rank of the Gain Ratio list and followed by the second last, the third last and so on. From the experiments, it was found that the accuracies were approximately equal (around 70%) from the number of selected attributes (17 to 13). The best accuracy for the model was found when the number of attributes was 12 (72.44%). The best result of the decision tree analysis is shown in Figure 5.14 and its accuracy is shown in Figure 5.15. If the number of attributes was reduced further, the trend of accuracy declined significantly up to around 64% (where the number of attributes was 8).

Table 5.7 shows various accuracy measurements obtained when the number of attributes were reduced form 17 to 7 and Figure 5.16 shows the trend of the J48 accuracy per number of attributes.

From the experiments, a number of rules were extracted, even though not all the rules were found to be interesting. Some of the interesting rules are summarised as follows:

Rule 1: IF (Tot_qz <= 0.74) AND (L8_scope = 2) THEN exam = Low

Rule 2: IF (Tot_qz <= 0.74) AND (L8_scope = 4) AND (n_lo <= 13) THEN exam

= Moderate

Rule 3: IF (Tot_qz <= 0.74) AND (L8_scope = 3) AND (Tot_HWL <= 0.49) THEN exam = Low

Rule 4: IF (Tot_qz <= 0.74) AND (L8_scope = 3) AND (Tot_HWL > 0.49) THEN exam = Moderate

Rule 5: IF (Tot_qz > 0.74) AND (Tot_HWL <= 0.96) THEN exam = Moderate

Rule 6: IF (Tot_qz > 0.96) THEN exam = High

Useful Information related to Learning Behaviour

The decision tree revealed some important information that were translated into rules. The result suggests that the total quizzes marks (Tot_qz) and week 8 lecture quiz mark about scope (L8_scope) played an important role in the students' success in the exam. It was found that the students who achieved less that 74% in the total of all quizzes would only be able to achieve "moderate" or "low" exam mark depended on the other assessment components such as lecture quiz in week 8 (L8_scope) and total home work/lab mark (Tot_HWL) (rule 1 to 4). On the other hand, if the students were able to achieve more than 74% of the total lecture quizzes then they would be able to obtain at least "moderate" exam mark (rule 5 and 6). This behaviour can be identified as a positive learning behaviour.

```
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

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

Tot_qz <= 0.74
|   L8_scope = 2.0: Low (3.0)
|   L8_scope = 4.0
|   |   n_lo <= 13: Moderate (2.0)
|   |   n_lo > 13: Low (4.0)
|   L8_scope = 5.0: Low (6.0/1.0)
|   L8_scope = 3.0
|   |   Tot_HWL <= 0.49: Low (3.0)
|   |   Tot_HWL > 0.49: Moderate (5.0/1.0)
|   L8_scope =
|   |   Tot_Lec_qz <= 0.35: Low (51.0/9.0)
|   |   Tot_Lec_qz > 0.35
|   |   |   n_task <= 9: Low (2.0)
|   |   |   n_task > 9: Moderate (5.0)
|   L8_scope = 7.0: Moderate (2.0)
|   L8_scope = 1.0: Moderate (2.0)
Tot_qz > 0.74
|   Tot_HWL <= 0.96: Moderate (59.0/16.0)
|   Tot_HWL > 0.96
|   |   Tot_qz <= 0.96: Moderate (2.0)
|   |   Tot_qz > 0.96: High (10.0)

Number of Leaves  :     14

Size of the tree :      22
```

Figure 5.14: The screen shoot of the best classification model

In pedagogical perspective, these rules contain valuable information that can be useful for the teacher or the course coordinator. Based on this information, the teacher could make a revision to the course plan, for example because it usefulness, the proportion of the lecture quizzes could be increased during the semester.

The results of the classification analysis on the combined dataset suggested a strong positive correlation between between the total quiz marks and the total homework/labs marks and the exam mark. The figures indicate very encouraging support for *the fourth hypothesis* about the correlation between quiz marks and exam and *the six hypotheses*

```
Number of Leaves  :     14

Size of the tree :      22


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         113               72.4359 %
Incorrectly Classified Instances        43               27.5641 %
Kappa statistic                          0.5304
Mean absolute error                      0.2425
Root mean squared error                  0.3792
Relative absolute error                 60.546  %
Root relative squared error             84.8134 %
Total Number of Instances              156
```

Figure 5.15: The best accuracy for the J48 decision tree model

about the correlation between total homework/labs marks and exam.

Table 5.7: A comparison of various accuracy measurements

No of Attribute	TP rate	Precision	Recall	Accuracy	Variable Removed
17	0.699	0.7	0.699	69.87	Example 0-3
16	0.699	0.7	0.699	69.87	HWLwk8
15	0.705	0.706	0.705	70.51	L5_ pointers
14	0.699	0.697	0.699	69.87	HWLwk12
13	0.699	0.699	0.699	69.87	HWL_attd
12	0.724	0.723	0.724	72.44	qz_attd
11	0.686	0.681	0.686	68.59	L8_scope
10	0.705	0.701	0.705	70.51	Lec_qz_ attd
9	0.673	0.674	0.673	67.31	HWLwk4
8	0.635	0.638	0.635	63.46	L2_mem_ diagram
7	0.66	0.665	0.66	66.03	Tot_lec_qz

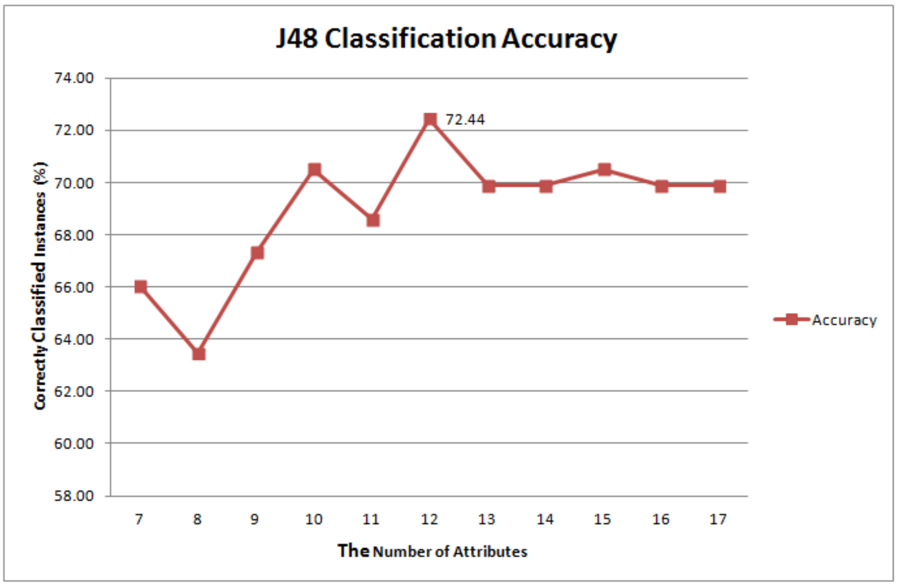


Figure 5.16: The trend of J48 accuracy per number of attributes selected

5.3.2 Association Rules Analysis

In this experiment, we utilised a combined categorical dataset discussed in Section 4.3.2 for the association rules mining analysis. Seven attributes were selected from Table 4.11 and Table 4.12 for the experiment. These attributes were the number of tasks (num_task) and the number of learning objectives (num_lo) performed in the Reflect system, the homework/lab mark (Total_HWLab), the quiz mark (Total_qz), the RFLT evidence (Rflt_cate), the total self-assessment (tSeflAssess) and the exam mark (exam_cate).

As discussed in Section 5.1.4, lift is a correlation measurement between two or more item-sets (attributes) and is measured by dividing the confidence by the support of that relationship. The lift coefficient less than 1 indicates that the occurrences of two item-sets are negatively correlated. If lift coefficient is greater than 1, then the occurrences of two item-sets are positively correlated and if the lift coefficient is equal to 1, then the occurrence of two item-sets are independent, meaning there are no relationships between these two item-sets. After the data attributes were selected, the Weka's Apriori algorithm with minimum confident 0.9 and lift correlation coefficient 1.1 was executed and the resulted output is shown in Figure 5.17.

Useful Information related to Learning Behaviour

A classification analysis using the Apriori algorithm was applied to the combined dataset in order to identify any associations of attributes that occurred together in the data. These associations and values are often represented by IF-THEN relationships. Our purpose was to look for attributes that, if they occurred together, would lead to a better exam mark. The algorithm was executed with a minimum confidence 0.9 and lift coefficient 1.1.

In regard to the results, although not all the rules discovered were interesting, some of them were quite promising. For example, rule number 2 stated that if a student studied large number of learning objective (n_lo) and obtained high total quiz (total_qz) mark, it would lead to a high total homework/lab mark (with 0.98 confidence and 1.94 lift). Furthermore, rule number 3 associated the number of learning objectives done in Reflect (num_lo) with total homework/labs marks (total_HWLab), leading to a high total_quiz mark (total_qz) (with 0.83 confidence and 1.82 lift).

```
Apriori
=======

Minimum support: 0.3 (47 instances)
Minimum metric <lift>: 1.1
Number of cycles performed: 14

Generated sets of large itemsets:

Size of set of large itemsets L(1): 13

Size of set of large itemsets L(2): 8

Size of set of large itemsets L(3): 1

Best rules found:

 1. total_HWLab=High 79 ==> num_lo=High total_qz=High 49    conf:(0.62) < lift:(1.94)>
 2. num_lo=High total_qz=High 50 ==> total_HWLab=High 49    conf:(0.98) < lift:(1.94)>
 3. num_lo=High total_HWLab=High 59 ==> total_qz=High 49    conf:(0.83) < lift:(1.82)>
 4. total_qz=High 71 ==> num_lo=High total_HWLab=High 49    conf:(0.69) < lift:(1.82)>
 5. total_HWLab=High 79 ==> total_qz=High 62    conf:(0.78) < lift:(1.72)> lev:(0.17)
 6. total_qz=High 71 ==> total_HWLab=High 62    conf:(0.87) < lift:(1.72)> lev:(0.17)
 7. num_lo=High 73 ==> total_HWLab=High total_qz=High 49    conf:(0.67) < lift:(1.69)>
 8. total_HWLab=High total_qz=High 62 ==> num_lo=High 49    conf:(0.79) < lift:(1.69)>
 9. num_lo=High 73 ==> total_HWLab=High 59    conf:(0.81) < lift:(1.6)> lev:(0.14) [22
10. total_HWLab=High 79 ==> num_lo=High 59    conf:(0.75) < lift:(1.6)> lev:(0.14) [22
```

Figure 5.17: Association rules using combined dataset

Besides a number of interesting rules, the Apriori algorithm also discovered some less convincing rules. For example, rule number 1 and number 10 were redundant (with confidence < 0.80).

These rules can be used by teachers to focus on the areas that lead to an improvement in the students' performance. The findings can also be shared with students in order to make them aware about what to expect and what learning aspects to focus on.

The results of the association mining analysis suggested that there is a strong positive correlation between the number of learning objective (num_lo) learnt in the Reflect system and the total_quiz mark (total_qz) or homework/labs (total_HWlab) marks. In the previous analyses, it has been shown that the total_quiz mark (total_qz) or homework/labs (total_HWlab) marks correlated positively to the exam mark. Therefore, achieving good total quiz mark (total_qz) or homework/labs (total_HWlab) marks or both would increase the chances to achieve a good exam mark. The figures indicate very encouraging support for *the fourth hypothesis* and *the fifth hypothesis* as that were discussed at the beginning of this section.

5.4 Conclusion

As reported in this chapter, we carried out a number of experiments to test the hypotheses. The statistical analyses were able to show correlations among the data attributes but could not distinguish the group of weak students from the group of strong ones. Therefore, various EDM methods namely clustering with K-means, classification with J48 and association rules mining with Apriori algorithm were applied to the Reflect dataset. As a result, the importance of self-assessment and the reflection in learning were revealed. The results suggested that there was a strong positive correlation between the number of self-assessment performed in the Reflect system and the exam marks as well as the total self-assessment performed in the system and the exam. These results were able to support the hypotheses related to the students' activity in the Reflect system.

The EDM methods such as classification and association rules were also employed to the students' assessment components. The results indicated that the total quiz marks and the homework/labs marks were correlated positively to the exam marks.These results provided additional support to the hypotheses proposed in the present study.

In this chapter, the deployment of the data mining techniques in order to explore the data was described and the findings were explained in order to extract meaningful and useful information and knowledge from the data. This meaningful information could be extracted from the data after a series of statistical and data mining experiments was performed. We argue that the findings from the experiments were able to prove the hypotheses and provided answers to the research questions.

Chapter 6

Conclusion and Future Work

6.1 Conclusions

The foremost objective of this thesis was to study how EDM methods can be applied to the dataset that come from a system that promotes learning by reflection. A large number of previous studies have suggested that reflection improves learning. The main purpose was to seek potentially interesting information and patterns inside the data. While a lot of EDM research studied about student's learning behaviours, the research that investigate the students' behaviours related to reflective learning remains unexplored.

In this study, our main objective is to evaluate the effectiveness of EDM methods for: (1) gaining knowledge about students' learning behaviours; (2) identifying which behavioural patterns lead to positive or negative outcomes; and (3) extracting knowledge about the impacts of reflection on learning.

Based on these study objectives, the research question and subsidiary research question have been defined in Chapter 1. In order to address the research question and subsidiary research question, we have conducted a study on students' learning behaviours when learning with a system that supports learner reflection (i.e., Reflect). The present study employed EDM methods to examine the impact of those behaviours on the students' performance.

In addition, five hypotheses were formulated to address the research question and subsidiary research question as follow:

Hypothesis 1: *Students who submitted RFLT evidence above the class average in* *the Reflect system are more likely to perform well in the exam.*

Hypothesis 2: *The total number of self-assessments performed in the Reflect system correlates positively to exam performance.*

Hypothesis 3: *Fewer submissions of the example solutions and RFLT evidence in the Reflect system would lead to poor exam performance.*

Hypothesis 4: *Students who attend and achieve good marks on all quizzes have a better chance at performing well in final exam.*

Hypothesis 5: *Students who miss weekly homework/lab exercises are at an increased of risk of performing poorly in the final exam.*

The main hypotheses (H1 to H3) are related directly to the use of Reflect system as a mean of reflection and were evaluated using the Reflect dataset. The other two hypotheses (H4 and H5) were developed and tested by using the students' assessment data from the course coordinator.

The present study employed three most popular educational data mining methods along with statistical analyses to prove the hypotheses and to address the research question and subsidiary research question as discussed in Chapter 5. The EDM methods and statistical analyses that were used to prove the hypotheses are:

- Statistical analyses utilising scatter plot and the Pearson correlation analysis were used to prove H1 and H2.

- Clustering utilising K-means algorithm was used to prove H3.

- Classification utilising J48 algorithm was used to prove H1, H2, H4 and H5.

- Association rules mining utilising Apriori algorithm was used to prove H1, H2, H4 and H5.

Beside the data gathered from the Reflect system (students' activity data), this study also used the students' assessment data obtained from the course coordinator and the combination of students' activity and students' assessment data. Based on the application of EDM methods to these three datasets, we have been able to achieved the following learning objective:

Gaining knowledge about students' learning behaviours

Identification of students' learning behaviour in particular those that can distinguish students, can provide important information and insights towards the students' performance. In this study, we have identified three groups of students distinguished by their participations in the self-assessment activities in the Reflect system. These activities were designed to promote self-learning by reflecting on their understanding of programming concepts and learning objectives set-up by the teacher in the Reflect system. The activities were exercising the tasks and learning objectives, writing and submitting programming codes and self-assessing the example solutions in the system.

From results of the present study, we were able to identify three groups of students who self-assess themselves in the Reflect system: the *regular users* who used the Reflect system for self-assessment frequently, the *moderate users* who performed self-assessment irregularly in the Reflect system, and the *casual users* who occasionally used the Reflect system for reflective learning. By applying diverse EDM methods to the Reflect data as discussed throughout the thesis, we were able to distinguished each group learning behaviours. The regular Reflect users were characterised by very active participation in exercising the self-assessment tasks and leaning objectives (Tot_self-assessment), viewing and exercising the example solutions (example solution evidence), and submitting their own programming codes (RFLT evidence) to the Reflect system. The moderate Reflect users were characterised by the moderate participations in self-assessment activities in the Reflect system. For example, they submitted on average medium number of tasks and learning objectives, and medium RFLT evidence submission. Meanwhile, the casual user characterised by fewer submissions in all submission categories).

Identifying which behavioural patterns lead to positive or negative outcomes

Using three different datasets, we were able to identify learning behaviour that lead to positive or negative outcomes (in the exam). From the exploration of students' activity (Reflect) dataset, we identified two learning objectives that lead to a positive outcome. First: the *students' determination* to submit their own correct solutions above the class average for the programming tasks that were recorded as RFLT evidence in the Reflect system (Section 3.4.1) and second: the *self-regulated learning* to exercise the learning objectives of a task (recorded as Tot_self-assessment evidence in the Reflect system) as discussed in Section 3.3.1. These were revealed by the results of experiments related to hypothesis 1 to 3 (H1 - H3) that showed a strong positive correlation between these learning behaviours to the exam performance.

In addition, the result of experiments related to H3 also revealed that less active group of students gave little commitment to exercising the self-assessment tasks and tended to submit fewer solutions of example solutions to the Reflect system. These behaviours could be associated with the learning behaviours that lead to negative outcomes.

From the exploration of students' assessment dataset and combined dataset, we identified *the students' consistency* in attending and solving the quiz problems (Tot_qz) as the learning behaviour that lead to a positive outcome (in the exam). This was revealed by the results EDM research to prove H4. In addition, the result of experiment related to H5 revealed that *the lack of commitment* to practice the weekly homework/labs exercises (Tot_HWL) could be identified as leaning behaviour that lead to a positive learning outcome.

Extracting knowledge about the impacts of reflection on learning

In this thesis, the reflection is defined as the process in which "individuals engage to explore their experiences in order to lead to a new understanding and appreciation" One way to perform reflection for learning is by doing a self-assessment in which a person reflecting on his/her ability or understanding of learning concepts and making judgment whether he/she have met certain criteria or standards.

The present study used the educational data gathered from an online system that supports learner reflection (i.e., the Reflect system) to gain new insights and knowledge into students' learning behaviours particularly that related to self-assessment. As a result, the information related to students' learning behaviour have been gathered during the study. This information was presented in Chapter 5 in the designated section called *"Useful Information related to Learning Behaviour"*. In this section, we would like to summarise this information as follows:

- Students with a strong commitment to success tried to work hard and self-regulated themselves to perform diverse self-assessment tasks in the Reflect system. These self-assessment tasks including self-assessing the tasks and learning objectives and comparing them against the criteria set-up by the teacher, submitting programming codes and example solutions to the Reflect system in order to compare it with the example solutions from the tutor. These reflective learning behaviours have positive impact toward the students performance (in the exam). The students' final exam marks indicated this impact. The the results of this study have shown a strong positive correlation between performing self-assessment tasks in the Reflect system and the exam mark.

- Self-assessment for learning is important because it can help students to develop the ability to identify their strengths and weaknesses and focus their study efforts on the particular area they believe needs improvement.

- Self-assessment is essential to students as it enables them to consider how effective their learning habits are against their current performance.

The information and knowledge resulting from this study can also be useful for teachers to better understand their students' learning behaviours and to inform students if their current behaviour is associated with negative or positive outcomes. The teacher, for example, may use the models that resulted from the classification analyses to classify new students and predict their final exam marks before the end of the semester. In addition, the resulting clustering information may be used to help students to form group collaborative work with only one student from each cluster. The information can also be used to identify students at risk; hence, interventions can be initiated to help them.

As the final conclusion, this study has able to achieve the research objectives and provide answers to the research question and subsidiary research question that have been formulated to provide direction for the study. As the objective of EDM research is to provide a deeper understanding of the key factors that impacting on learning. This study have achieved this by investigating students' learning behaviour, in particular how students perform self-assessment and reflect on their learning behaviour in ways that affect the learning outcomes.

6.2 Study Limitations and Future Work

A number of issues emerged during the present study either related to the data or the interpretation of the results. First, the present study only used the Reflect data gathered in one semester; hence, the analyses were limited. The results would be more interesting if the data mining was conducted for data that had been collected for many years. Second, the data was not designed in the first place to be suitable for mining. As a result, the data was complex, contained a lot of noise, and was heterogeneous. On one hand, the data was unique because it was designed to tailor different users' preferences and knowledge background. On the other hand, the uniqueness of the data made it difficult to mine. Third, most of the educators involved in designing a Web-based educational system and evaluating students' learning activities in the system do not have sound knowledge of using data mining techniques. Therefore, a majority of teachers and tutors may not be familiar with EDM and may find it difficult to set up the system to generate the data that is ready for mining.

The main purpose of the application of data mining in education is to find patterns and gain insights about students' performance that may be used by educators to help students in learning. Hence, this objective is subjective and difficult to measure compared to the application of data mining to other areas such as e-commerce where the objective can easily be measured, for example, by the increased amount of money received by a company or the increased number of customers.

Bibliography

[1] Agrawal, R., and Srikant, R. Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. Very Large Data Bases, VLDB* (12-15 1994), J. B. Bocca, M. Jarke, and C. Zaniolo, Eds., Morgan Kaufmann, pp. 487–499.

[2] Allevato, A., Thornton, M., Edwards, S. H., and Perez, M. A. Mining data from an automated grading and testing system by adding rich reporting capabilities. In *Proc. Educational Data Mining Conference (EDM08)* (Quebec Canada, 2008), pp. 167–176.

[3] Baker, R. *Data mining for education in International Encyclopedia of Education*. 3rd Ed. Elsevier, Oxford, UK.

[4] Baker, R. S. Modeling and understanding students' off-task behavior in intelligent tutoring systems. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2007), ACM, pp. 1059–1068.

[5] Baker, R. S., Corbett, A. T., Koedinger, K. R., and Wagner, A. Z. Off-task behavior in the cognitive tutor classroom: when students "game the system". In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2004), ACM, pp. 383–390.

[6] Baker, R. S., and Yacef, K. The state of educational data mining in 2009:a review and future vision. *Journal of Educational Data Mining 1*, 1 (2009).

[7] Baruque, C. B., Amaral, M. A., Barcellos, A., Jo a. C. d. S. F., and Longo, C. J. Analysing users' access logs in moodle to improve e learning. In *EATIS '07: Proceedings of the 2007 Euro American conference on Telematics and information systems* (New York, NY, USA, 2007), ACM, pp. 1–4.

[8] Bayer, J., Bydzovská, H., Géryk, J., Obšivac, T., and Popelinskỳ, L. Predicting drop-out from social behaviour of students. In *Proceedings of the 5th International Conference on Educational Data Mining-EDM 2012* (2012), pp. 103–109.

[9] Beck, J. E., and Mostow, J. How who should practice: Using learning decomposition to evaluate the efficacy of different types of practice for different types of students. In *Intelligent Tutoring Systems* (2008), pp. 353–362.

[10] Benchaffai, M., Debord, G., Merceron, A., and Yacef, K. Tada-ed, a tool to visualize and mine students' online work. In *Proceedings of International Conference on Computers in Education, (ICCE04)* (Melbourne, Australia, 2004), E. McKay and B. Collis, Eds., RMIT, pp. 1891–1897.

[11] Black, E. W., Dawson, K., and Priem, J. Data for free: Using lms activity logs to measure community in online courses. In *Internet and Higher Education* (2008).

[12] Bouchet, F., Azevedo, R., Kinnebrew, J. S., and Biswas, G. Identifying students characteristic learning behaviors in an intelligent tutoring system fostering self-regulated learning. In *Proceedings of the 5th International Conference on Educational Data Mining* (2012), pp. 65–72.

[13] Boud, D. Implementing student self-assessment. In *Campbelltown: Higher education research and development society of Australasia (HERDSA Green Guide)* (1991), vol. 5.

[14] Boud, D., and Keogh, R. Promoting reflection in learning: A model. In *Reflection: turning experince into learning* (1985), D. Boud, R. Keogh, and D. Walker, Eds., pp. 18–40.

[15] Byrne, P., and Lyons, G. The effect of student attributes on success in programming. In *ITiCSE '01: Proceedings of the 6th annual conference on Innovation and technology in computer science education* (New York, NY, USA, 2001), ACM, pp. 49–52.

[16] Castro, F., Àngela Nebot, and Mugica, F. Extraction of logical rules to describe students' learning behavior. In *WBED'07: Proceedings of the sixth conference on IASTED International Conference Web-Based Education* (Anaheim, CA, USA, 2007), ACTA Press, pp. 164–169.

[17] Castro, F., Vellido, A., Nebot, A., and Mugica, F. Applying data mining techniques to e-learning problems. 2007, pp. 183–221.

[18] Cocea, M. Assessment of motivation in online learning environments. In *Adaptive Hypermedia and Adaptive Web-Based Systems* (2006), Springer, pp. 414–418.

[19] Cocea, M., Hershkovitz, A., and de Baker, R. S. J. The impact of off-task and gaming behaviors on learning: Immediate or aggregate? In *AIED* (2009), pp. 507–514.

[20] Del Prette, Z. A. P., Del Prette, A., De Oliveira, L. A., Gresham, F. M., and Vance, M. J. Role of social performance in predicting learning problems: Prediction of risk using logistic regression analysis. *School Psychology International 33*, 6 (2012), 615–630.

[21] Dringus, L., and Ellis, T. Using data mining as a strategy for assessing asynchronous discussion forums. *Computer & Education 45* (2005), 141–160.

[22] Dunham, M. H. *Data Mining: Introductory and Advanced Topics*. Upper Saddle River NJ, 2003.

[23] Dweck, C. S., and Elliott, E. S. Achievement motivation. *Handbook of child psychology 4* (1983), 643–691.

[24] Eccles, J., and Wigfield, A. Teacher expectations and student motivation. *Teacher expectancies* (1985), 185–226.

[25] Fayyad, U. M. Data mining and knowledge discovery: Making sense out of data. *IEEE Expert: Intelligent Systems and Their Applications 11*, 5 (1996), 20–25.

[26] Feng, M., Beck, J., Heffernan, N., and Koedinger, K. Can an intelligent tutoring system predict math proficiency as well as a standardized test. In *Proc. Educational Data Mining Conference (EDM08)* (2008).

[27] Gallier, J. H. Unification procedures in automated deduction methods based on matings: A survey. In *Tree Automata and Languages*. 1992, pp. 439–.

[28] Garcia, E., Romero, C., Ventura, S., and Calders, T. Drawbacks and solutions of applying association rule mining in learning management systems. In *Proc. EC-TEL07 Workshop on Applying Data Mining in e-Learning (ADML07)* (2007).

[29] Hagan, D., and Markham, S. Does it help to have some programming experience before beginning a computing degree program? In *ITiCSE '00: Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSEconference on Innovation and technology in computer science education* (New York, NY, USA, 2000), ACM, pp. 25–28.

[30] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. The weka data mining software: an update. *SIGKDD Explor. Newsl. 11*, 1 (Nov. 2009), 10–18.

[31] Hammouda, K., and Kamel, M. *Data mining in e-learning.* Advanced Information and Knowledge Processing. Springer, 2006, pp. 1–28.

[32] Han, J., and Kamber, M. *Data Mining Concept and Techniques*, 3rd ed ed. Morgan Kaufman, San Fransisco, USA, 2012.

[33] Hanna, M. Data mining in the e-learning domain. *Campus-Wide Information Systems 21*, 1 (2004), 29–34.

[34] Jeong, H., and Biswas, G. Mining student behavior models in learning-by-teaching environments. In *Proceeding of the 1st International Conference on Educational Data Mining* (2008), pp. 127–136.

[35] Kay, J., and Li, L. Scrutable learner modelling and learner reflection in student self-assessment. In *Intelligent Tutoring Systems: Proceedings 8th International Conference, ITS 2006* (2006), M. Ikeda, K. Ashley, and T. Chan, Eds., Springer, pp. 197–206.

[36] Kay, J., Li, L., and Fekete, A. Learner reflection in student self-assessment. In *Proceedings of ACE 2007, 9th Australasian Computing Education Conference* (Sydney, 2007), Australian Computer Society, pp. 89–95.

[37] Kay, J., Maisonneuve, N., Yacef, K., and Reimann, P. The big five and visualisations of team work activity. In *Intelligent Tutoring Systems: Proceedings 8th International Conference, ITS 2006* (2006), Springer, pp. 197–206.

[38] Kay, J., Maisonneuve, N., Yacef, K., and Zaiane, O. Mining patterns of events in students' teamwork data. In *Online Proceedings of the ITS (Intelligent Tutoring Systems) 2006 Workshop on Educational Data Mining* (Jhongli, Taiwan, 2006), pp. 45–52.

[39] Klassen, K. J., and Smith, W. Web log analysis: a study of instructor evaluations done online. *Journal of Information Technology Education 3* (2004), 291–312.

[40] Koutri, M., Avouris, N., and Daskalaki, S. *A survey on web usage mining techniques for web-based adaptive hypermedia systems*. IRM Press, 2005, pp. 125–149.

[41] Lopes, C. T., and David, G. Higher education web information system usage analysis with a data warehouse. In *Computational science and its applications (ICCSA)* (Berlin, 2006), Springer.

[42] Lopez, M., Luna, J., Romero, C., and Ventura, S. Classification via clustering for predicting final marks based on student participation in forums. In *Proceedings of the 5th International Conference on Educatiional Data Mining* (2012), pp. 148–151.

[43] Lowes, S., Lin, P., and Wang, Y. Studying the effectiveness of the discussion forum in online professional development courses. *Journal of Interactive Online Learning 6*, 3 (2007), 181–210.

[44] Lu, Y., and Bettine, J. Data mining: an experimental undergraduate course. *J. Comput. Small Coll. 18*, 3 (2003), 81–86.

[45] Ma, Y., Liu, B., Wong, C. K., Yu, P. S., and Lee, S. M. Targeting the right students using data mining. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2000), ACM, pp. 457–464.

[46] Maehr, M. L., and Midgley, C. Enhancing student motivation: A school-wide approach. *Educational Psychologist 26*, 3-4 (1991), 399–427.

[47] Matthews, M., and Mitrovic, T. Do students who see more concepts in an its learn more? In *Proc. Educational Data Mining Conference (EDM08)* (Quebec Canada, 2008).

[48] Mazza, R., and Milani, C. Exploring usage analysis in learning system: gaining insights from visualisations. In *12th International conference on artificial intelligence in education (AIED 2005)* (Amsterdam The Netherlands, 2005).

[49] Mazzolini, M., and Maddison, S. When to jump in: the role of the instructor in online discussion forums. *Computers & Education*, 49 (2007), 193–213.

[50] McCuaig, J., and Baldwin, J. Identifying successful learners from interaction behaviour. In *Proceedings of the 5th International Conference on Educational Data Mining* (2012), pp. 160–163.

[51] Merceron, A., and Yacef, K. A web-based tutoring tool with mining facilities to improve learning and teaching. In *Proceedings of 11th International Conference on Artificial Intelligence in Education (AIED03)* (Sydney, 2003), F. Verdejo and U. Hoppe, Eds., IOS Press, pp. 201–208.

[52] Merceron, A., and Yacef, K. Educational data mining: a case study. In *Proceedings of the 12th Conference on Artificial Intelligence in Education* (Amsterdam, The Netherlands, 2005), C. K. Looi, G. McCalla, B. Bredeweg, and J. Breuker, Eds., IOS Press, pp. 467–474.

[53] Merceron, A., and Yacef, K. Tada-ed for educational data mining. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning 7*, 1 (2005).

[54] Merceron, A., and Yacef, K. Interestingness measures fo association rules in educational data. In *Proc. Educational Data Mining Conference (EDM08)* (Mar. 2008).

[55] MERRIAM, S. *Qualitative Research and Case Study Applications in Education. Revised and Expanded from "Case Study Research in Education."*. Jossey-Bass Publishers, San Fransisco, USA, 1998.

[56] MILTIADOU, M., AND SAVENYE, W. C. Applying social cognitive constructs of motivation to enhance student success in online distance education. *AACE Journal 11*, 1 (2003), 78–95.

[57] MONK, D. Using data mining for e-learning decision making. *The Electronic Journal of e-Learning 3*, 1 (2005), 41–54.

[58] MOODLE. http://moodle.org. last consulted 1 feb 2009.

[59] MOSTOW, J. Some useful design tactics for mining its data. ITS.

[60] MOSTOW, J., AND BECK, J. Some useful tactics to modify and map and mine data from intelligent tutors. *Natural Language Engineering 12*, 2 (2006), 195–208.

[61] MOSTOW, J., AND ZHANG, X. Analytical comparison of three methods to evaluste tutorial behaviours. In *Proc. Educational Data Mining Conference (EDM08)* (Quebec Canada, 2008), pp. 28–37.

[62] PAHL, C. *Data Mining for the Analysis of Content Interaction in Web-based Learning and Training Systems*. WIT Press, 2006, pp. 41–56.

[63] PECHENIZKIY, M., CALDERS, T., VASILYEVA, E., AND BRA, P. D. Mining the student assessment data: lessons drawn form a small scale case study. In *Proc. Educational Data Mining Conference (EDM08)* (2008), pp. 187–191.

[64] PERERA, D., KAY, J., KOPRINSKA, I., YACEF, K., AND ZAÏANE, O. R. Clustering and sequential pattern mining of online collaborative learning data. *IEEE Trans. on Knowl. and Data Eng. 21*, 6 (2009), 759–772.

[65] PERERA, D., KAY, J., YACEF, K., AND KOPRINSKA, I. Mining learners' traces from an online collaboration tool. In *Proceedings of Educational Data Mining workshop* (2007), pp. 60–69.

[66] PSOTKA, J., MASSEY, L. D., AND MUTTER, S. A. *Intelligent tutoring systems: Lessons learned*. Psychology Press, 1988.

[67] Quinlan, R. J. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, January 1993.

[68] Romero, C., and Ventura, S. *Data Mining in E-Learning*. Advances in Management Information. WIT Press, 2006.

[69] Romero, C., and Ventura, S. Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications 33*, 1 (Jul 2007), 135–146.

[70] Romero, C., Ventura, S., and Bra, P. Knowledge discovery with genetic programming for providing feedback to courseware authors. *User Modeling and User-Adapted Interaction 14*, 5 (January 2004), 425–464.

[71] Romero, C., Ventura, S., Bra, P. D., and Castro, C. D. Discovering prediction rules in aha! courses. In *User Modeling (UM 2003)* (Berlin Heidelberg, 2003), P. B. et al., Ed., Springer-Verlag, pp. 25–34.

[72] Romero, C., Ventura, S., de Castro, C., Hall, W., and Ng, M. H. Using genetic algorithms for data mining in web-based educational hypermedia systems. In *Adaptive Hypermedia and Adaptive Web-Based Systems (AH02)* (Málaga (Spain), 2002).

[73] Romero, C., Ventura, S., Delgado, J. A., and De Bra, P. Personalized links recommendation based on data mining in adaptive educational hypermedia systems. 2007, pp. 292–306.

[74] Romero, C., Ventura, S., Espejo, P., and Hervas, C. Data mining algorithms to classify students. In *Proc. Educational Data Mining Conference (EDM08)* (Quebec Canada, 2008), pp. 8–17.

[75] Romero, C., Ventura, S., and Garcia, E. Data mining in course management systems: Moodle case study and tutorial. *Computers & Education In Press, Corrected Proof* (2007).

[76] Sabourin, J. L., Mott, B. W., and Lester, J. C. Early prediction of student self-regulation strategies by combining multiple models. *differences 1* (2012), 4.

[77] Schon, D. A. *The reflective practitioner*. Basic Books. New York, 1983.

[78] SCHON, D. A. *Educating the reflective practitioner*. Jossey-Bass Publishers, San Fransisco, 1987.

[79] SHEARD, J., CEDDIA, J., AND HURST, G. Inferring student learning behaviour from website interactions: A usage analysis. *Education and Information Technologies 8*, 3 (2003), 245–266.

[80] SHEARD, J., CEDDIA, J., HURST, J., AND TUOVINEN, J. Inferring student learning behaviour from website interactions: A usage analysis. *Education and Information Technologies 8*, 3 (2003), 245–266.

[81] SHIH, B., KOEDINGER, K., AND SCHEINES, R. A response time model for bottom-out hints as worked examples. In *Proc. Educational Data Mining Conference (EDM08)* (Quebec Canada, 2008), pp. 117–126.

[82] SRIKANT, R., AND AGRAWAL, R. Mining generalized association rules. *Future Generation Computer Systems 13* (Nov 1997), 161–180.

[83] STIPEK, D., FEILER, R., DANIELS, D., AND MILBURN, S. Effects of different instructional approaches on young children's achievement and motivation. *Child Development 66*, 1 (1995), 209–223.

[84] TANG, T., AND MCCALLA, G. Smart recommendation for an evolving e-learning system: Architecture and experiment. *International Journal on e-Learning 4*, 1 (2005), 105–129.

[85] TANIMOTO, S. Improving the prospects for educational data mining. In *Complete On-Line Proceedings of the Workshop on Data Mining for User Modeling, at the 11th International Conference on User Modeling* (2007), pp. 106–110.

[86] URBANCIC, T., SKRJANC, M., AND FLACH, P. Web-based analysis of data mining and decision support education. *AI Commun. 15*, 4 (2002), 199–204.

[87] VENTURA, S., ROMERO, C., AND HERVAS, C. Analyzing rule evaluation measures with educational datasets: a framework to help the teacher. In *Proc. Educational Data Mining Conference (EDM08)* (Quebec Canada, 2008), pp. 177–181.

[88] Wang, Y., and Beck, J. E. Using student modeling to estimate student knowledge retention. *International Educational Data Mining Society* (2012).

[89] Witten, I. H., and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed ed. Morgan Kaufmann, San Fransisco, USA, October 2011.

[90] Zaiane, O. R. Web usage mining for a better web-based learning environment. In *Proceedings of Conferen on Advanced Technology for Education* (2001), pp. 60–64.

[91] Zorrilla, M. E., Menasalvas, E., Marin, D., Mora, E., and Segovia, J. Web usage mining project for improving web-based learning sites. In *10th International conference on computer ainded systems theory* (Las Palmas de Gran Canaria Spain, 2005).