University of the Witwatersrand

Educational Data Mining (EDM) in a South African University: A Longitudinal Study of Factors that Affect the Academic Performance of Computer Science I students

Lebogang Mashiloane
Student no. 0707701a

Degree of Master of Science by research only:

A Dissertation submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science.

Supervisor: Mr. Mike Mchunu

Signed on September 10, 2015 in Johannesburg

DECLARATION

I declare that this Dissertation is my own, unaided work. It is being submitted for the Degree of Master of Science at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.

Signature of candidate

Date

ABSTRACT

The past few years have seen an increase in the number of first year students registering in the School of Computer Science at Wits University. These students come from different backgrounds both academically and socially. As do many other institutions, Wits University collects and stores vast amounts of data about the students they enrol and teach. However this data is not always used after being stored. The area of Educational Data Mining (EDM) focuses on using this stored data to find trends and patterns that could enhance the knowledge about the student's behavior, their academic performance and the learning environment.

This longitudinal study focuses on the application of EDM techniques to obtain a better understanding of some of the factors that influence the academic performance of first year computer science students at the University of the Witwatersrand. Knowledge obtained using these techniques could assist in increasing the number of students who complete their studies successfully and identifying students who are at risk of failing and ensuring that early intervention processes can be put into place. A modified version of the CRISP-DM (CRoss-Industry Standard Process for Data Mining) was used, with three data mining techniques, namely: Classification, Clustering and Association Rule Mining. Three algorithms were compared in the first two techniques while only one algorithm was used in the Association Rule Mining. For the classification technique, the three algorithms that were compared were the J48 Classifier, Decision Table and Naïve Bayes algorithm. The clustering algorithms used included the Simple K-means, Expectation Maximization (EM) and the Farthest First algorithm. Finally, the Predictive Apriori algorithm was selected as the Association Rule Mining technique.

Historical Computer Science I data, from 2006 to 2011, was used as the training data. This set of data was used to find relationships within the data that could assist with predictive modeling. For each of the selected techniques a model was created using the training data set. These models were incorporated in a tool, the Success or Failure Determiner (SOFD), that was created specifically as part of this research. Thereafter, the test data set was put through the SOFD tool in the testing phase. Test data sets usually contain a variable whose value is predicted using the models built during the training phase. The 2012 Computer Science I data instances were used during the testing phase. The investigations brought forth both expected and interesting results. A good relationship was found between academic performance in Computer Science and three of the factors investigated: Mathematics I, mid-year mark and the module perceived to be the most difficult in the course. The relationship between Mathematics and Computer Science was expected, However, the other two factors (mid-year mark and most difficult module) are new, and may need to be further investigated in other courses or in future studies. An interesting finding from the Mathematics investigation was the better relationship between Computer Science and Algebra rather than Calculus. Using these three factors to predict Computer Science performance could assist in improving throughput and retention rates by identifying students at risk of failing, before they write their final examinations. The Association Rule Mining technique assisted in identifying the selection of courses that could yield the best academic performance overall, in first year. This finding is important, since the information obtained could be used during the registration process to assist students in making the correct decisions when selecting the courses they would like to do. The overall results show that using data mining techniques and historical data collected at Wits University about first year Computer Science (`CS-1`) students can assist in obtaining meaningful information and knowledge, from which a better un-

derstanding of present and future generations of `CS-1` students can be derived, and solutions found to some of the academic problems and challenges facing them. Additionally this can assist in obtaining a better understanding of the students and factors that influence their academic performance. This study can be extended to include more courses within Wits University and other higher educational institutions.

Keywords. Educational Data Mining, CRISP-DM, Classification, Clustering, Association Rule Mining, J48 Classifier, Decision Table, Naïve Bayes, Simple K-means, Expectation Maximization, Farthest First, Predictive Apriori

ACKNOWLEDGEMENTS

Firstly I would like to thank God for granting me the strength to continue with this process till the end. Secondly I would like to thank my supervisor, Mr. Mike Mchunu, for his advice and guidance throughout the research and writing process. I would also like to acknowledge Ms. Daisy Matlou (Senior tutor, School of Mining Engineering) and Ms. Nontokozo Mpofu (formerly Associate Lecturer, School of Computer Science), two members of the MSc panel set up by the School of Computer Science to assess, review and provide feedback on the proposal that eventually led to this dissertation. My gratitude is also passed to Ms. Sabrina Liccardo and Ms. Bindu Cherian for their assistance in the proofreading and editing of the document. Lastly, thanks to my husband, Landi Mashiloane, and family for the consistent support and love.

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background of the Problem

Throughput in higher education is vital for the development of any country. South Africa's democratization has affected different sectors of society in different ways. One of these sectors, the education sector, has undergone a lot of transformation in the past few years. The Outcome-Based Education (OBE) system was introduced in South African schools in 1998 by the former Minister of Education, the late Professor Kader Asmal [Pasensie 2010]. It was later discovered that the OBE system was not efficient. As a result, a Revised National Curriculum Statement (RNCS) was introduced in 2002. Following from this, a 10-point programme, better known as The Roadmap, became South Africa's ruling party's educational manifesto in 2009 [Pasensie 2010]. The transformation of the educational system was also driven by a number of factors, such as:

- race and gender equity policies,
- the changing institutional structures of universities,
- curriculum, teaching and research imperatives,
- globalization and internationalization

Following from the educational changes that were introduced in 2009 and the implementation of the National Bench-marking Tests (NBTs), there was an increase in the grade 12 pass rate. For instance the pass rate increased from 60.6% in 2009 to 73.9% in 2012. The new system included curriculum changes implemented by the Department of Education. This impacted the higher education sector, resulting in an influx of students both applying to and entering university for the first time.

Another factor that contributed to the large number of first-year university entries was the South African government's provision of financial and material support to needy, but academically capable students. This has enabled them to access university education, making it possible for them to proceed towards completion of their studies. Although the funding these students receive may be insufficient in meeting all their needs, the student financial aid scheme does go some way towards ensuring that deserving and capable students are not deprived of a university education on the grounds of poverty. Every year, the amount spent by government on education is increased by about 5% [MacGregor 2009]. In the recent 2010/11 national budget, a total amount of R900.9 billion was allocated for overall expenditure [Treasury 2010]. Of this amount, R165.1 billion was allocated to support and fund education (see Figure 1.1). Approximately 18%, the largest allocation of the total budget, is directed towards education. This amount is followed closely by allocations towards the economic affairs and social protection departments. Of the total amount of money allocated to education the higher education sector, which constitutes the broad

scope within which this research is being performed, has been allocated a significant portion of the government's budget.

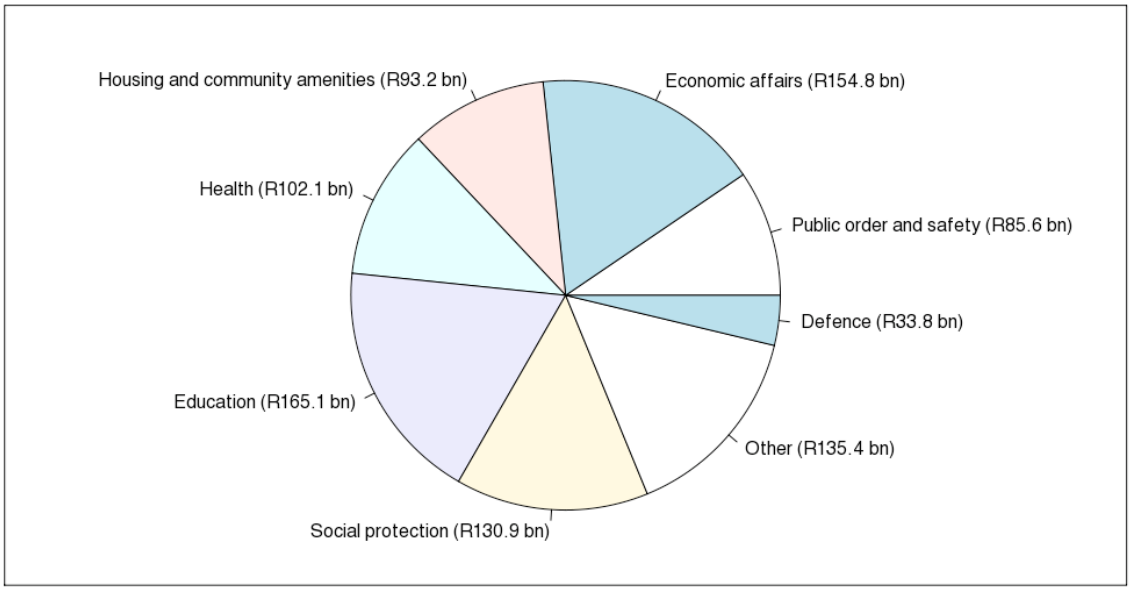


Figure 1.1: 2010-2011 South African Government Expenditure Budget

The University of the Witwatersrand (hereinafter referred to as Wits University), where this research was performed, has benefited from the influx of first-year students to tertiary institutions. An increasing number of students, particularly first-year students, are enrolling at Wits University. First-year student applications and enrolments have continued to grow since 2009. What is also clearly noticeable about these students is their diversity. They come from different academic, social, racial, religious and economic backgrounds. In short, they are a microcosm of the South African population itself.

The increasing enrolment of students in South African higher education institutions is to be welcomed. However, this trend is not matched by the hoped-for success in terms of performance, retention and graduation rates. The graduation, retention and success rates remain low. The negative impact resulting from these problems goes beyond mere academic interest. Students who, for one reason or another, do not complete their degrees on time or fail to complete their degrees constitute a financial burden on the country, more specifically the South African taxpayer. If this problem were to be looked at from a business perspective, it would be fair to say that the money invested by government and other stakeholders in the higher education enterprise has not yet begun to yield the expected "profits" or "dividends".

Higher education institutions collect vast amounts of data about students, most of which is hardly ever used. There is a need for institutions to be more informed and knowledgeable about their students, and for them to understand some of the reasons behind these students' academic performance and behaviour. One of the ways in which this can be done is for such institutions to obtain information and knowledge about their students by mining, processing and analysing the vast amount of data they accumulate about them. Student-related data contains many aspects that could provide meaningful information to institutions about their students, information that could lead to a better understanding of who these students really are. Institutions could use this understanding to put in place mechanisms and processes aimed, among other things, at improving students' academic life, identifying students who are most likely to fail or drop out, and intervening timeously to prevent these students from dropping out or failing. The large repository of data that institutions maintain about their students can be processed and analysed to find answers to some of the most important questions that arise in relation to these students.

This research uses a relatively new approach, Educational Data Mining (EDM), to try and answer some of the questions currently facing higher education institutions about the nature of the students they admit. There are questions concerning a number of different issues. For instance, what factors influence the academic performance of students [Superby *et al.* 2006; Cheewaprakokbit 2013]?, which are the most important amongst the factors used to predict academic performance [Affendey *et al.* 2010]? Is it possible to obtain information from institutional data about the types of students who are most likely to be "at risk" [Rienks and Taylor 2009]? Can we derive a better understanding of students by viewing them as members of clusters characterized by a similarity in attributes, such as race, gender and age? In many disciplines, data mining has been and continues to be used to obtain information from large data sets. Given the vast amount of data generated and stored by higher education institutions such as Wits University, data mining techniques have begun to be used as part of the research carried out in the educational sector. As a consequence a new research area has emerged, in the form of Educational Data Mining (EDM). Baker and others [2010] define EDM as a scientific area that uses data to get a better understanding of students and and their learning environment.

This work investigates data mining techniques and their use in educational contexts. More specifically, the research focuses on the study and analysis of a subset of the large amount of data collected by different entities at Wits University. In particular first year Computer Science I (`COMS1000`) students data was used. This data was obtained from the Student Enrollment Centre (SENC), the Faculty of Science and the School of Computer Science. The data contains several attributes of interest. These include student gender, race, selected academic courses, performance, and other important details. Selected data mining techniques were used to answer the questions posed in this study.

1.2 Problem Statement

The past few years have seen an increase, both locally and internationally, in the number of students entering higher education institutions. The more students enrol, the more data is collected and stored in institutional repositories such as databases, data marts and data warehouses. Despite the fact that vast amounts of student-related data is being collected and stored by many higher education institutions, not much of it gets used in any meaningful way, either to obtain information or to generate new knowledge from it. These repositories contain different kinds of data about students, such as their personal (student number, age, race, gender, residential address, etc.) and academic performance data. All this data could be used in a meaningful way, for example, to improve and develop the teaching and learning activities of academics within their departments or schools [Vranic *et al.* 2007].

According to Siraj and Abdoulha [2009], "universities are data-rich but information poor". This is a direct consequence of the universities collecting and archiving data, without using it to obtain meaningful information. This research looks at how data mining techniques can be used to obtain information from a subset of the vast amount of data stored at Wits University. At Wits University the primary data set is maintained and managed globally, at university level. A level below this are the different faculties, with each faculty having the responsibility to collect and manage its own data. The Faculty of Science controls nine schools, amongst which is the School of Computer Science. The school collects and manages its own data. However, as and when permitted to do so, the school is able to obtain data from other entities in the university, such as the Faculty of Science, as well as other schools. Each school in the university, including the School of Computer Science, accepts hundreds of students every year.

Between 2006 and 2011 the School of Computer Science registered an estimated total of 635 first year students, about 127 students annually. These are the `COMS1000` or `CS-1` students. During this 5-year period the school has collected, recorded and stored some data about each student. The data includes the following details: student number, first and last name, units and modules registered for, and academic

performance (pass or fail). At the end of the academic year and once the examination results have been discussed, verified and/or amended by a meeting of the Faculty of Science's Board of Examiners (BOE), all the data is kept by the faculty and some of it is stored in the School of Computer Science. Apart from being used to answer only simple queries, the data is thereafter not used for much else. At Wits University data is captured and stored in the Oracle data warehouse. Using the Oracle Discoverer tool, the Academic Information and Systems Unit (AISU) queries the database to obtain information, in the form of simple reports. These reports do not include any information predicted from the data. Neither do they include information from which different patterns and trends present within the data can be identified. To obtain such useful information, different techniques of handling the data are required.

Despite the fact that many South African higher education institutions, including Wits University, amass lots of data within their environments, not many of them have begun to leverage the power of data mining and data mining techniques to process and analyse this data, and to use the retrieved information to support their planning and decision-making processes. Furthermore, not much evidence exists to suggest that educational researchers in most South African universities, including Wits University, have begun to look at data mining, more specifically Educational Data Mining (EDM), as a possible area of research they can engage in, amongst others, to get to know more about their students, their characteristics, the environment in which they learn and in which they are taught. By focusing on data mining and the use of data mining techniques in educational contexts, more specifically at Wits University and within the School of Computer Science, this research aims to contribute some knowledge to the currently "sparse" knowledge base on research being conducted in Educational Data Mining (EDM) at Wits university, faculty and school.

With EDM having recently become an established area of research, it has now become possible to mine available data to obtain useful information from it. Such information can provide knowledge that may enable institutions to develop a better understanding of certain aspects of the teaching and learning environment. This includes understanding the types of students being taught, their strengths and weaknesses, analysing and predicting their performance, and how best to teach the courses or revise the curricula to benefit the students. The knowledge obtained from mining the data could lead to changes that could benefit the students, school, faculty and university. EDM now affords us the opportunity to use the large amount of unused data to try and find answers to some of the unanswered questions. It is clear from this discussion that if educational institutions continue to archive and not use the large data sets at their disposal, they are depriving themselves of a major source of useful information and knowledge about their environments.

The past few years have seen a steady increase in the number of first year students who come to Wits University to study Computer Science. These students come from diverse academic, social, economic and ethnic backgrounds. A minority of them will have done a computer programming subject at high school. Depending on the high school academic environment and the outlook on the importance of mathematics, some of the students will have done mathematics at an advanced level. Typical computer science classes are also characterized by a clear gender imbalance, in which males predominate over females. The School of Computer Science is no exception in this regard. The students also come from different types of high schools. Some of them come from expensive, independent private schools and colleges where the standard of education, as well as the qualifications and experience of their teachers, are very high. Another group of students are products of former model C schools, which enjoy reasonably good facilities and in which well-qualified teachers ensure that high educational standards and good outcomes continue to be maintained. Yet another group of students, the majority, obtain their education in African township and rural high schools. Most of these high schools are characterized by poor facilities, inadequate resources and generally high failure rates. Consequently, some of the students from these schools, given their poor academic background and inadequate English-language ability, tend to struggle in their studies, and do not perform as well as expected [Rauchas *et al.* 2006]. Results from statistical analysis

of the language questionnaires presented in Nel and Müller [2010] have shown that limited teacher proficiency in English has a negative impact on students' English language proficiency and learning.

According to the literature , there is an increase in the number of students who are unprepared to enter college [Gabriel 2008]. In the South African context, Strydom *et al.* [2010] speak of how institutions have little to no control of the educational preparation of the students they enrol. These students stand the highest risk of dropping out, de-registering, being excluded, or failing to complete their degrees in record time, if at all. The purpose of this research is to see how data mining methods, applied within the context of the School of Computer Science, can provide us with a better and deeper understanding of the diverse classes of first year Computer Science students we teach in the school. One of the things this research aims to enable the school to do, using classification data mining methods, is to predict the final, end-of-year marks in the `COMS1000` unit, using the mid year results obtained in two `COMS1000` modules in the first semester. Classification is a data mining technique that uses a classification model to predict the class to which an object (data instance) belongs. For this purpose `COMS1000` datasets, collected between 2006 and 2011, were used. Given the importance of Mathematics in the Computer Science curriculum, this research focused on the relationship between Mathematics I and Computer Science I, as well as on the influence of Mathematics I on performance in Computer Science I. The research also focused on cluster analysis, in which the `COMS1000` dataset was segmented into clusters in order to identify groups of students with similar characteristics. Cluster analysis is a data mining technique that assigns objects to groups (clusters), based on their similarity. Objects in the same cluster are more similar to one another compared to objects in other clusters. Having this knowledge will enable the school to address the specific needs of different groups of students in the CS-I class. Association rule mining (ARM) discovers patterns or associations between items in a dataset. The results are presented in a form that is easy to understand. In this research ARM was applied to the `COMS1000` dataset to discover patterns in the data.

1.3 Significance of the Study

This research is motivated by the need to obtain a better understanding of the first year students who are taught in the School of Computer Science. Higher education institutions such as Wits University collect vast quantities of data, from which useful information can be obtained through data mining techniques. From this data, knowledge can be obtained that could be used to enhance the education of students and the teaching and learning practice. The School of Computer Science collects a lot of data about its students. The bulk of this data is not used or analysed in any way to obtain useful information about these students, more specifically the first year students. The `COMS1000` data used in this research was collected from the School of Computer Science as well as the Faculty of Science. It was collected during the period 2006-2010. To obtain information from it, different data mining operations were used. Classification methods will enable us to predict, for example, whether certain students will succeed or are at risk of dropping out or failing. Armed with this knowledge, the School of Computer Science is then better placed to take remedial steps to try and improve the chances of success for at-risk students. Clustering methods will enable us to identify groups of students who share similar characteristics, on the basis of which provision can be made to cater for their specific needs. Using Association Rule Mining techniques will also make it possible to uncover hidden relationships between some of the data items in the present repositories.

1.4 Research Question

The main question, which this research sought to answer, is as follows:

> *Can the historical data collected at Wits University about first year Computer Science (CS-1) students be used to obtain meaningful information and knowledge, from which a better understanding of present and future generations of CS-1 students can be derived, and solutions found to some of the academic problems and challenges facing them*?

In order to answer this question, the following sub-questions were posed:

1. *In what way do the factors of race and gender influence performance in a diverse Computer Science I classroom*?
2. *What is the impact of a difficult module on overall performance in Computer Science I*?
3. *How effective and accurate are classification methods as predictors of overall performance in Computer Science I, when applied on the first semester performance in the first two Computer Science I modules?*
4. *What is the nature of the relationship between Mathematics I and Computer Science I, and their respective modules?*
5. *How strong is the association between the courses selected during registration by Computer Science I students and their overall academic performance?*
6. *What is the relationship between Computer Science I and overall performance in the first year of the BSc. degree?*

1.5 Research Methods

The course content in the Computer Science I topics has for a number of years remained the same. The same four modules (Basic Computer Organization (BCO), Data and Data Structures (DDS), Fundamental Algorithmic Concepts (FAC) and Limits of Computation (LOC)) have been taught to generations of Computer Science I students. Over the years, a lot of data about these modules has been collected and stored. The fact that data mining techniques are now being increasingly used in educational settings opens the opportunity to retrieve, process, analyze and obtain meaningful information from the data about these courses, data that has been lying dormant in the university repositories for a number of years. It was decided that a longitudinal study would be the the most appropriate research method to follow, given the nature of this research and the data that would be used to help answer the research questions posed earlier. In a longitudinal study, change is measured over a period of time [Lewis-Beck *et al.* 2003]. Such a study may involve cohort analysis, where the same group is investigated over several years or it may involve the same variable (independent of the sample group) being measured over a period of time [Lewis-Beck *et al.* 2003]. In this study we focus on the latter approach, given the fact that the same variables were being studied, about which data has been collected repeatedly from year to year, for a number of years. The longitudinal aspect of this study is similar to the work of Mortagy and Boghikian-Whitby [2010] and Pollock [2009], who studied and analyzed, over a period of time, student performance in specific courses.

The main purpose of mining the data about the first-year Computer Science students is to obtain information that could be used, among other things, to assist in finding ways of improving student throughput, alert students, lecturers, tutors, and course coordinators to situations where poor academic progress could result in failure, de-registration or dropping out. This research addresses a number of important issues

related to first year Computer Science students who study at Wits University. It focuses on several important problems, including demographic and academic ones.

In order to find answers to some of the questions in this research, different data mining techniques were used. Obtaining information from the variables or factors under study could assist in answering some of these questions. Selected classification techniques were used to obtain information about the Computer Science I module considered or perceived by students to be the most difficult. To predict performance in Computer Science I, the marks obtained in Mathematics I and the first semester marks of two Computer Science I modules were used. The Computer Science I mark was used to determine whether it could possibly be used as a predictor of first-year performance. The race and gender factors were used to study their influence among different clusters of CS-I students and to determine their effect on performance. Finally, the Association Rule Mining technique was used to determine whether the relationships amongst the first year courses selected by individual students contributed to their success or failure in the first year of their degree studies.

The aim in following the data mining approach was to investigate these factors within the context of a relatively new and emerging research discipline; namely Educational Data Mining (EDM). Given the fact that EDM is a relatively young and promising research discipline and that, despite the presence of large volumes of student-related and other institutional data, there is not much evidence of work being done at Wits University, as well as other South African universities, which focusses on research in EDM. By conducting and locating this research within this new, exciting and promising discipline, this work attempts to contribute towards filling this gap. It is hoped that within the School of Computer Science, faculty of Science and University of the Witwatersrand, and perhaps in other South African universities, there will be a keen interest by academic researchers to take a new look at the vast amount of student-related data at their disposal, to see how data mining techniques can be used to leverage this data by transforming it into useful and meaningful information. These are some of the reasons the Educational Data Mining approach was taken in conducting this research.

Three data mining techniques were selected and used for modelling: Classification, Clustering and Association Rule Mining (ARM). During this phase several classification, clustering and association rule models were created, using a selection of open-source data mining algorithms implemented in the WEKA toolkit (the acronym WEKA stands for "Waikato Environment for Knowledge Analysis") [Hall *et al.* 2009]. Three classification methods (J48, Decision Table and Naïve Bayes) were selected and their performance compared using the same dataset. For clustering a similar approach was taken, with three clustering algorithms (Simple K-means, Expectation Maximization (EM) and the Farthest First) being used and their performance compared with each other. To generate association rules, the A priori algorithm was used.

A total of 826 instances belonging to students registered in Computer Science I between 2006 and 2012 were used. Of these, 713 (86%) instances collected between 2006 and 2011 were used for training the selected classifiers. The remaining 113 instances (13%), collected in 2012, were used as the test dataset. The reason for splitting the data in this manner, by using a large training set and a comparatively smaller test set, was to obtain better accuracy in the results and to minimize bias and variance effects. Using the the most rcent year as the test data and the rest of the years as the training set was done to maintain the idea of historical data predicting future instances. A limitation of this is that it sometimes creates a bias. Furthermore the 10-fold cross validation used when building the models, allowed for the training of the data 10 times. This is explained in detail later in the dissertation.

CRISP-DM (CRoss-Industry Standard Process for Data Mining) is a standard process model used by data mining practitioners in developing projects. The model is used as part of a project's development life cycle [Chapman *et al.* 2000]. There are six phases in CRISP-DM: business understanding, data un-

derstanding, data preparation, modelling, evaluation and deployment. CRISP-DM was adopted as part of the methodology to be followed in this research. A dedicated software tool, named "Success or Failure Determiner" (SOFD), was designed and implemented for use during the modelling, evaluation and deployment phases of the CRISP-DM process model.

1.6 Contributions of this Dissertation

Educational Data Mining is an emerging research discipline. EDM techniques are used to extract meaningful information from the data collected by educational institutional institutions. Not much literature or information is readily available that points to EDM techniques being widely adopted or used by the South African educational research community. In this section, we highlight the contributions that have been made by the work we have done in this research.

1.6.1 Contribution to EDM Research within the School of Computer Science

As far as we know, none of the faculties or schools at Wits University is engaged in research or studies in which data mining techniques are used to extract meaningful information or to generate new knowledge from educational datasets. Within the School of Computer Science, this work contributes immensely towards strengthening the foundation that has been built to initiate research in EDM.

Additionally, the School of Computer Science currently does not possess any tool that can be used by lecturers, tutors and course coordinators to obtain meaningful profiles about the first year students they interact with. Such a tool could help, for example, to identify students who risk failure or who are likely to drop out. Furthermore, there is no tool available that can be used to recommend courses individual CS-I students must take in order to improve their chances of success in first year. In order to address this shortcoming another contribution has been made, in the form of a software tool that can be used to perform prediction, clustering and association rule mining tasks. In recognition of the tool's relevance, we were invited to demonstrate it at the 6th International Conference on Educational Data Mining (EDM 2013) in Memphis, Tennessee.

1.6.2 First Semester Marks and Performance in the COMS1000 Unit

From this work the following facts have emerged:

- The overall `COMS1000` final mark can be predicted using the marks obtained from the modules written in the first semester.
- Over the years, the failure rate in the Limits of Computation (LoC) module has been higher compared to the other three `COMS1000` modules. For this reason and for the purpose of this research, LoC has been considered to be a "difficult" module. Of interest in this study was to find out whether performance in this module could be used to predict overall performance in the `COMS1000` module. In this regard, the work has shown that CS-I students who do well in the LoC module also tend to do well in the `COMS1000` unit.

These findings contribute to a better understanding of the `COMS1000` students and the potential academic difficulties some of them are likely to be confronted with. Having this information makes it possible for the school to take proactive steps to prevent these students from dropping out or failing the `COMS1000` unit.

1.6.3 The Relationship between Computer Science I and Mathematics I

Mathematics plays an important role in the Computer Science curriculum offered at Wits University. Computer Science I students are required to take a compulsory course in Mathematics I (Algebra and Calculus). They are not permitted to progress to Computer Science II unless they have passed Mathematics I. It was therefore interesting and important to study the relationship between Mathematics I and Computer Science I. In the Wits University context, Mathematics I was found to have a strong and significant correlation with Computer Science I. Significant correlations were also found between the two Mathematics I modules (Algebra and Calculus) and the majority of the `COMS1000` modules. These findings contribute to a better and more informed understanding of the role and influence of Mathematics I on performance in Computer Science I. For example, poor performance or high failure rates in Computer Science I can be addressed by looking at the students' performance in Mathematics I.

1.6.4 Contribution to EDM Research by South African Higher Education Institutions

In addition to the specific contributions mentioned earlier, this study contributes a data mining perspective and dimension to the overall research being conducted at South African higher education institutions. Most of these institutions are plagued by the familiar problems of students dropping out, failing, de-registering and being excluded. The adoption of EDM as an area of research may enable researchers in different institutions to know more about their students, including having the ability to profile them. For example, using EDM techniques, the researchers will be able to identify potential failures or dropouts. Being aware of these and other problematic cases, the education community is then better placed to focus on intervention programmes. Not only can this help the students, it can also enable higher education institutions to increase their throughput rates. Additionally, during the time this work was being done, efforts were also being made to publicize some of its findings. Please see Appendix K for the list of publications accepted during the process of conducting this research.

1.7 Structure of the Dissertation

Chapter 2 is the Literature Review chapter of this dissertation. It provides a detailed discussion of the topics relevant to this research with related work. Chapter 3 gives a brief background of the work with key concepts used in the dissertation. It goes on to discuss the research method that was followed in conducting this research and the chapter concludes with a detailed description of the approach that was followed, which is based on the CRISP-DM process, a standard approach that is used in data mining projects.

Chapter 4 presents and discusses the results obtained in this research. The results were used to provide answers to the research questions and their related sub-questions. Chapter 5 summarizes the significance of this research. Chapter 6 presents the limitations experienced during the course of this work and discusses future work that can result from this research.

Chapter 2

Literature Review

2.1 Introduction

Large amounts of data are collected by educational institutions. Some of this data, after being collected and archived, is never processed and analysed to obtain information or generate knowledge from it. Thanks to computer technology it is now possible to store the collected data in databases, data warehouses and other repositories. However, using databases and other repositories only addresses the problem of data collection and storage, not the issue of data analysis. Data mining and data mining techniques make it possible to address this problem. Data mining techniques are used, among other things, to discover knowledge from data [Du 2010]. From its inception data mining has grown to encompass a number of different fields, in which useful applications are being implemented.

The purpose of this chapter is to describe and discuss literature that is related to the current research. The related work is integrated in a way so that it flows from an overview of data mining to the specific problems addressed in this dissertation. This chapter is structured as follows: Section 2.2 briefly introduces and discusses the concept of data mining. It is then followed by a discussion of the different applications of data mining in Section 2.3. Section 2.4 then focuses specifically on Educational data mining, and discusses some of the literature and work related to this research. Thereafter, Section 2.5 provides a rationale for using data mining to address the research problem. From a data mining perspective this research focused on three techniques: classification, clustering and Association Rule Mining. For each technique data mining algorithms were selected and used in the various experiments that had to be done. The final section, Section 2.6 looks at these selected techniques and methods.

2.2 Data Mining

The term *data mining* is used to refer to the integral part of the Knowledge Discovery process that finds and reveals hidden information from raw data [Du 2010; Merceron and Yacef 2005]. Organizations and institutions collect and store vast amounts of data in different repositories. This may be historical or operational data. Using data mining methods, the data can be processed to reveal meaningful patterns and information useful for decision-making purposes. Data mining is a fast-growing area, with applications being implemented in different sectors. Not all sectors are at par in terms of their growth, in some sectors data mining has not been explored as yet. And in others, it is still emerging as a field of interest. This section is a brief overview of data mining and its techniques and applications.

The tasks or functions involved in data mining are generally classified into two categories: predictive and descriptive [Velickov and Solomatine 2000; Sanderson 2008]. With prediction, data models are constructed and used to predict unknown values, after the models have been trained using data with known values. Description uses models that identify "human-interpretable patterns, associations or correlations

describing the data" [Velickov and Solomatine 2000]. Prediction and description involve different data mining operations, as illustrated in Figure 2.1.

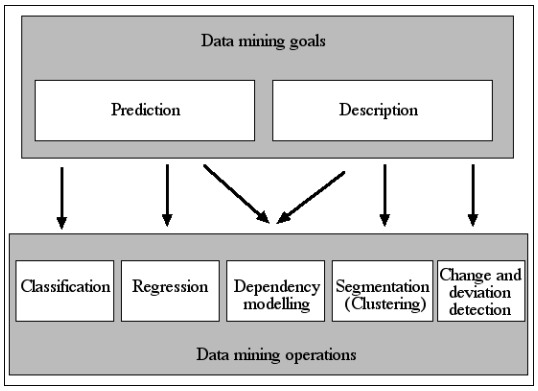


Figure 2.1: Data Mining Goals and Operations [Velickov and Solomatine 2000]

For each operation, Figure 2.2 illustrates some of the data mining techniques that can be used to perform that operation.

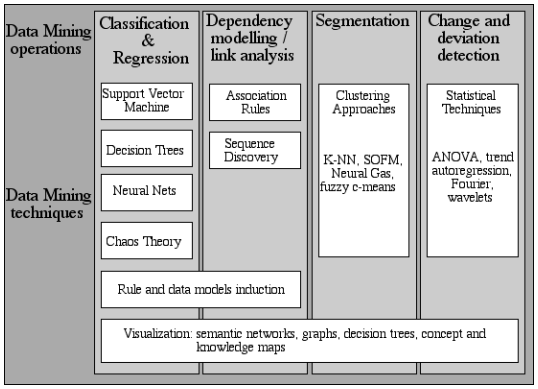


Figure 2.2: Data Mining Operations and Techniques [Velickov and Solomatine 2000]

Proprietary and open source tools have been developed to perform different data mining tasks. For example, some classification software is produced by Salford systems, and SAS provides clustering and segmentation tools. For this research the WEKA tool kit was used as a back-end to the GUI-based software that was developed as a frontend (Success Or Failure Determiner tool). WEKA is a machine-learning open source tool developed by the University of Waikato in Hamilton, New Zealand. WEKA is an acronym for *Waikato Environment for Knowledge Analysis*. The tool provides a variety of machine-learning techniques, some of which were used in this research.

Three data mining operations were selected and used in this study: classification, clustering and Association Rule Mining. From each operation different techniques were used to perform experiments on the datasets that were collected for this study.

2.3 Applications of Data mining

2.3.1 Data Mining in Business and Commerce

The use of data mining methods in business and commerce has been extensively researched and explored. In both sectors data mining methods and applications have been used the longest. This is a well-researched area, and many papers have been written on the use of data mining techniques in business. According to Ghani and Soares [2006], almost all business processes involve some form of data mining. Customer relationship management, supply chain optimization and business intelligence are only a few of the business processes that have come about as a result of business engaging in data mining [Ghani and Soares 2006]. Workshops and conferences are also being held to highlight the importance of the role played by data mining in the areas of business and commerce. An example where data mining could be used is in predicting or forecasting the final price of an item in an auction [Jank *et al.* 2006]. Within the business sector data mining methods are also used to find patterns in daily transactions. The information obtained may be used by a business to generate ad-hoc decisions, to ensure survival in a competitive environment [Zaïane 1999].

Despite the fact that data mining has yielded great results in business and commerce, challenges and pitfalls still remain. For example, there is the issue of discontinuous dialogue between the research community and business practitioners [Ghani and Soares 2006]. Another is the issue of privacy, which makes it difficult for most businesses to provide their data freely for researchers to use [Agrawal and Srikant 2000]. According to Kohavi *et al.* [2002], even though business people may be experts in their respective fields, they usually lack expertise in data analysis. For this reason, there is still a need for data mining tools and expertise to be used in business.

2.3.2 Data Mining in Science

The field of science encompasses many areas and disciplines such as medicine, biology, drugs and pharmaceutics, bioinformatics and so on. According to Zaki *et al.* [2002], the relatively new area of bioinformatics "provides opportunities for developing novel data mining methods." Data mining in science and more specifically, in bioinformatics, has shown tremendous progress in the past few years. Data mining methods are also used in designing drugs, a specialized field in scientific research. Some researchers support using classifiers, to identify positive cases in drug discovery and to reduce costs [Zaki *et al.* 2002].

2.4 Educational Data Mining

The benefits of data mining have begun to filter through to other disciplines. In education, a new research discipline has emerged, in the form of educational data mining (EDM). Some educational institutions have begun to use data mining techniques to discover useful patterns or trends in the data, and to support institutional decision-making processes. EDM covers many aspects of educational research, particularly research in the higher education sector. It focuses on both the teaching and learning in any institution. With related work ranging from academic performance predictions, grouping students according to behaviour and even associating the courses the students register for [Yadav and Pal 2012; Salazar *et al.* 2004; Kumar and Chadha 2012]. The work in educational data mining has not been limited to modelling the students currently studying. There has also been a reflection of the institution using historical data. For example, Merceron and Yacef [2005] discuss the use of data mining techniques to predict alumni most likely to donate to their institution. Related work has looked at monitoring student success and retention rates [Osmanbegović and Suljić 2012]. Among other things, EDM gathers all historical data in educational institutions, finds useful patterns of information and generates new knowledge. In their

work, Romero *et al.* [2008] describe how analysing large educational data sets can contribute towards an understanding of student behaviour. This can largely assist in enhancing the teaching and learning environment.

Educational data mining has also been linked to enabling instructors to improve their teaching [Merceron and Yacef 2005]. According to Merceron and Yacef [2005], teachers can use findings from data mining research to reflect on their teaching and also to re-design their courses or course materials. EDM techniques are also used to detect information patterns in data. In higher education, they can be used to identify students who are likely to drop out and to detect graduation patterns. Knowing all this information can enable institutions to come up with measures that would prevent students from dropping out or failing.

2.5 Justification of use of Data Mining for the Research Problem

According to Baker and Yacef [2009], research in EDM has been exclusive to only a few parts of the world, such as North America, Western Europe, Australia and New Zealand. It would seem that elsewhere, in places such as South Africa for instance, EDM research has not yet taken root. It is for this reason that it was felt, given the context and circumstances, EDM is an area worth exploring for this research. Wits university collects a large amount of data about different entities such as students, academic and administrative staff, alumni, and different kinds of service providers. Of primary interest in this research is the use of data mining techniques to analyse the data accumulated in the School of Computer Science and Faculty of Science, about the first year Computer Science students. One of the aims of this research was to see how data mining techniques, when applied to the university data sets, could be used to produce meaningful information about these students, information that could be used to develop a better, more informed understanding of their characteristics and behaviour, as well as the environment in which they learn.

The research problem is focused on understanding first year Computer Science students at Wits University. Literature has shown how educational data mining and its techniques can assist with the understanding of students [Romero *et al.* 2008]. Furthermore, there is a large amount of related work that uses data mining techniques to answer some of the research questions which have been deduced from the main research question.

The relationship between race and gender on the one hand and academic performance, on the other, is a much-studied topic. Different studies relating to these two factors have produced different outcomes. Most studies include these two factors amongst a number of factors under investigation [Salazar *et al.* 2004; Siraj and Abdoulha 2009]. Most of these studies also focus more on gender rather than race. For historical reasons, both factors are important in the South African context. We were therefore keen to investigate the role played by race and gender in the academic performance of Computer Science I students. Although the relationship between race and/or gender with academic performance has been investigated, there seems to be no research that focuses on just these two variables and uses cluster analysis. It may seem clear what the groupings are for race and gender, however an interest of the research was to identify the academic performance classified to those groupings. Salazar *et al.* [2004] discuss how clustering algorithms can be used to group students according to their academic performance. A study by Siraj and Abdoulha [2009] used cluster analysis to group students using gender, faculty and student status (enrol, move, expel, quit and completed) as the variables. In Science, Arts and Medicine there were more male students expelled compared to females, who successfully completed their studies in these disciplines. Male students were more successful in Sports and Law, where more of them completed their studies, compared to female students. An understanding of whether or not a particular race and gender grouping has any relation to academic success is an interesting one.

Using data mining to perform predictive analysis is largely common. Although in this research the factors used for prediction were selected and known, their ability to be used to correctly classify the academic performance of the Computer Science first year students was of interest. The ability of these factors to correctly predict the majority of the instances would lean towards a positive correlation and/or relationship between the factors observed was unknown. Furthermore, a good relationship between these factors would allow for some of the selected data mining classifiers to be included and used in the SOFD tool. The first model to be included in the tool was built using the data collected about the Computer Science I module perceived to be the most difficult.

Some of the courses that are taught in within the school contain one or more modules that are perceived to be difficult by students, course coordinators and lecturers. There are a number of reasons why a module or course can be perceived as difficult. These include the number of students who fail, the introduction of new and challenging content, and having to cope with a high workload in a course. In the School of Computer Science, the Limits of Computation (LoC) module is perceived to be the most difficult of the four modules taught in first year Computer Science. The LoC module was modified when a new first year Computer Science curriculum was introduced after 1998 [Sanders and Mueller 2000]. The curriculum changes were motivated, amongst other things, by the growing performance gap between students who had prior programming experience and those who did not. Students without prior programming experience tend to get demotivated. According to Sanders and Mueller [2000] it is important to focus on basic principles in the first-year curriculum, in order to prepare students for subsequent years. This research looked into some of the factors responsible for the perceived difficulty of the LoC module. Is academic performance in this module directly related to academic performance in Computer Science I? Wiedenbeck *et al.* [2004] tried to find factors that affect outcome in a programming course, while Jian *et al.* [2008] investigated how students handle difficulty in course work. No literature could be found which describes how performance in a module perceived to be difficult can be used to predict academic performance in a course containing that module. In tackling this question, we hope to initiate some work in this regard.

The purpose of the second model was to predict the final Computer Science I results, using the mid-year results (obtained from two Computer Science I modules that were done in the first semester). There are not many studies that contain information or details on how mid-year or first semester performance can be used to predict performance in a course at the end of the academic year. In a study conducted at a Mid-Western American university the classification technique was used to determine factors of success in a first-year Computer Science class [Campbell and McCabe 1984]. And in another study the academic performance of students during the first half of a semester [O'Byrne *et al.* 2009] was used to predict the final semester result. The authors were interested in the relationship between the in-semester marks and the final semester mark. They concluded that academic performance during a semester could not on its own be used to predict the final semester mark, and hence could not be used to predict students at risk of failure. This is not the same as approach taken in answering question 2 in this research, however is similar and therefore used as a guideline.

The study by Yadav and Pal [2012] used classification techniques to predict performance amongst engineering students. The aim of their study was to identify poor performing students and to assist them in improving their results. The following factors were studied: student academic history, number of children in the student's family and the student's admission type. Three decision tree algorithms were used: C4.5, ID3 and CART. The C4.5 algorithm emerged as the most accurate of three. According to the authors, algorithms such as the decision tree algorithm are effective in learning from prior data and using such data to create predictive models [Yadav and Pal 2012]. This finding lends support to the decision to use classification techniques to identify, among other things, students who are most likely to fail. This study is also amongst the many studies that support comparison of algorithms when investigating certain

variables. One of the purposes of the SOFD tool (created in this research) is to help in identifying these students.

Previous work in Computer Science Education (CSE) has identified and investigated the relationship between Mathematics on one hand, and Computer Science, information technology and programming courses, on the other hand. This is one of the relationships that was studied and analyzed in this research. Given the role of Mathematics in the Computer Science curriculum taught at Wits University, it was considered important to investigate the relationship between Mathematics I and Computer Science I as well as the influence, if any, of Mathematics I on performance in Computer Science I. Furthermore, it was also important and necessary to determine the correlation between the performance scores in Mathematics I and Computer Science I, including the correlation between their respective modules. White and Sivitanides [2003] conducted a study using data collected from 837 university students enrolled in a Computer Information Systems Programming course. The aim was to investigate the correlation between Mathematics and programming courses such as the Visual Programming course. A significant correlation was found between freshman Mathematics, SAT (Scholastic Aptitude Test) scores and Visual Programming [White and Sivitanides 2003]. In this research statistical methods were also used to analyze the relationship between Mathematics I and Computer Science I. The relationship between the respective modules of these two courses was also studied. The study by Eid and Millham [2012] revealed that students who obtained at least 65% in Calculus did better in any chosen IT course compared to those who obtained less than 65% [Eid and Millham 2012]. These findings confirmed the hypothesis made in the study about good performance in Mathematics (specifically Calculus) being a predictor of success in IT-related subjects.

At Wits university students who are not registered on a fixed curriculum are allowed to select any combination of courses they would like to do. After selecting their major subjects these students must take additional subjects to make up the number of points required to pass in a particular year of study. One of the aims of this study was to explore and analyze how the selection and combination of courses taken in conjunction with Mathematics I and Computer Science I by first year Computer Science students related to their performance in first year. The idea was to explore the performance implication of selecting a particular combination of courses together with Computer Science I as a major subject. A study whose subjects were Computer Science freshmen was conducted in the Facultad de Informatica, Valencia, Spain Fornés *et al.* [2008]. It was discovered that most students who passed Programming, Calculus, Physics and Computer Technology did not drop out of the degree they were doing. The results of this research can be viewed as contributing towards a better understanding of how different course combinations relate to or impact academic performance. According to Bhullar and Kaur [2012], one of the difficulties faced by educational institutions is to predict a path that individual students can follow during their studies. Penn-Edwards [2004] conducted a study at Griffith University, which determined whether different course combinations had an influence on academic performance in the English course. There was little difference in the average marks obtained using different course combinations, which led the authors to conclude that course combination had no effect on performance in the English course. In this study the Association Rule Mining (ARM) technique was used to determine the strength of the association between the courses selected by Computer Science I students and to see how this association related to overall performance in the first year of study.

The study by Aher and Lobo [2012] used Association Rule Mining to look at the issue of course selection by students. The aim was to create a tool that could be used to recommend courses that students could select. Similarly to our study, Aher and Lobo [2012] also used the WEKA toolkit, including its Association Rule Mining algorithm. The algorithm was applied on student records obtained from the departments of Information Technology, Computer Science & Engineering. Four types of Association Rule Mining algorithms were used: Apriori, Predictive Apriori, Tertuis Association Rule and Filtered Associator. The Predictive Apriori algorithm, which was also used in our research, produced the best

rule, which stated that students who were not interested in the Switching Theory & Logic Design and Operating Systems courses would be interested to do the Data Structures course. Since Aher and Lobo [2012] required an Association Rule Mining algorithm that would only produce "yes" rules, the Apriori algorithm was used as part of their recommender system.

One of the aims of this research was to determine whether the Computer Science I mark could be used to predict overall academic performance in first year. Researchers have investigated various factors to determine which amongst them can be used as predictors of student performance in first-year studies. Osmanbegović and Suljić [2012] carried out a study whose aim was to model students' academic success. The study used data collected over a 2-year period (2010-2011) from the University of Tuzla's Faculty of Economics. Various factors were investigated and different data mining methods were used to predict the academic success of first year students. Similarly to our work, the study also used WEKA's Naïve Bayes and C4.5 classification algorithms. The Naïve Bayes algorithm outperformed the other classifiers as a predictor of student performance. Kovacic [2010] used data mining techniques to predict academic performance. A total of 453 Information Systems students at the Open Polytechnic of New Zealand took part in the study, which explored factors such as gender, age, ethnicity, enrollment data and course programme. The study followed the CRISP-DM model, a modified version of which was used in this research. Four classification trees were generated and used for prediction. Ethnicity, course programme and course block had the most influence as predictors of overall success. The study also showed that enrollment data could not on its own be considered to be a performance predictor.

2.6 Selected Data Mining Methods for the Research Problem

Data mining involves a number of different tasks. These include description, estimation, prediction, classification, clustering and association [Larose 2014]. In this research each task was performed using different techniques. The following tasks were selected for study: classification, clustering, association rule mining, correlation analysis and regression analysis. Subsection 2.6.1 provides a detailed discussion of the different classification methods that were selected and used in this study. Clustering techniques are discussed in Subsection 2.6.2. This is followed, in Subsection 2.6.3, by a discussion of Association Rule Mining, Finally, correlation analysis and regression analysis are discussed in Subsection 2.6.4.

2.6.1 Classification

According to Baker and Yacef [2009], the different activities in Educational Data Mining (EDM) include prediction, clustering, relationship mining, distillation of data for human judgement and discovery with models. Classification uses supervised learning methods to predict class labels for data instances with unknown class labels [Zaïane 1999]. Classification produces a model by applying a supervised learning algorithm on a dataset, normally referred to as the *training set*. Once the model is created, it can be used to predict a value for a variable whose value is unknown or missing. Classification involves a training phase and a prediction phase [Sanderson 2008].

- In the training phase, data records whose class variables contain known values are used to create a classification model. These records make up the training set.
- In the prediction phase, a test set consisting of data records whose class variable values are unknown, is supplied to the model created during the training phase. The model then predicts the value of the class variable in each record.

Classification techniques are commonly used in educational research, where they play a significant role. For example, they are used to predict student performance [Merceron and Yacef 2005]. Some of these

techniques are implemented in the WEKA toolkit. Most of them use a training dataset, which a learning algorithm uses to build the required model [Zaïane 1999].

In their study Lodhi *et al.* [2011] compared three classification algorithms to determine the best predictor of the final grade in a database course. The Decision Tree, Decision Table and Radial basis function (RBF) Network classifiers were used. Of the three classifiers the Decision Tree and RBF Network each achieved a 70% success rate, with Decision Table only managing to achieve a 30% success rate. The authors selected the Decision Tree classifier as most suitable for processing and analysing the type of data they were using. Three classification algorithms were used in our study: J48, Decision Table and Naïve Bayes, all of which are available in the WEKA tool kit. These classifiers are described in greater detail in the following subsections.

J48 Classifier

This algorithm is one of several decision tree algorithms. It is the WEKA version of the C4.5 algorithm [Durant and Smith 2004]. J48 is purported to be accurate and quick in building models, and the results are presented in a simple-to-understand fashion [Bhullar and Kaur 2012]. The algorithm calculates the gain ratio for each attribute and thereafter creates a decision tree [Yadav and Pal 2012]. As shown in Yadav and Pal [2012], the tree is constructed by using the attribute with the maximum gain ratio as the root node. Pruning is used to remove unnecessary branches. Figure 2.3 illustrates a decision tree.

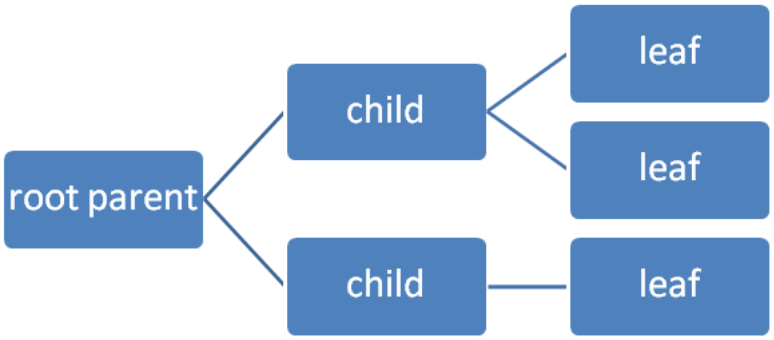


Figure 2.3: An example of a Decision Tree

A decision tree algorithm accepts the entire set of data and starts off with a root node (see Figure 2.3). Thereafter the dataset is split using a split scoring function [Nowozin 2012]. There are different types of split scoring functions. In this research the Information Gain function was used. Nowozin [2012] explains information gain as the mutual information between the left and right node with the predictive output. If the information gain (I) between two arbitrary variables a and b is calculated using left (L) and right (R), then the calculation is performed as follows [Nowozin 2012]:

$$I(a,b) = D_{KL}(p(a,b)|p(a)p(b))$$

Information gain is used to split the dataset into two children starting from the root. The process continues until the leaf node is reached [Myatt and Johnson 2009].

Decision Table

Decision tables are similar to decision trees, the difference being that their outputs are rules, instead of trees. Durant and Smith [2004] state that decision tables are simpler to use and that the decision table algorithm is computationally less demanding. The cross validation of decision tables is performed incrementally, which makes the process faster [Durant and Smith 2004]. The algorithm works by finding an attribute or combination of attributes that can best predict the class of a given data instance [Durant and Smith 2004]. It uses forward selection, which involves a top-to-bottom search in which attributes are added as the search proceeds [Durant and Smith 2004].

A decision table is a two-dimensional structure that represents a set of rules. Each rule consists of a *premise* (condition) and a *conclusion* (action). A decision table can therefore be viewed as representing condition-action rules, using a general structure similar to Table 2.1 (adapted from Arentze *et al.* [2008]).

Problem Area	
Condition Set	Condition Entries
Action Set	Action Entries

Table 2.1: Decision Table General Structure

The "Problem Area" section indicates the type of problem being investigated [Arentze *et al.* 2008]. The table is divided into four sections (quadrants) separated horizontally and vertically by double lines. The horizontal double line divides the table into an upper condition part and a lower action part. The vertical double line separates sets (condition and action) from entries (condition and action). The *condition set* is the set of attributes that make up the antecedents of the rules applied during the classification process [Arentze *et al.* 2008; Lima *et al.* 2009]. Each *condition entry* is a set of values an attribute in the condition set may have. These values occupy different columns in the condition entries section of the table. Every column specifies a classification rule. The set of possible actions a classifier may execute are specified in the *action set* component. An *action entry* contains the values of the corresponding action set [Lima *et al.* 2009].

Naïve Bayesian Network

The Naïve Bayesian Network (Naïve Bayes) is so-called because it assumes that the attribute values being used are independent [Pandey and Pal 2011]. Its advantages include its ability to scan the training data only once and being able to use only a small amount of this data to estimate the parameters [Pandey and Pal 2011]. A Naïve Bayes classifier is based on Bayes' formula, which uses the idea of conditional probability. Equation 2.1 shows Bayes' formula for calculating conditional probabilities [Riesenfeld 2011].

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.1}$$

where

- $P(A)$ is the prior probability of an event A.
- $P(B)$ is the prior probability of an event B.
- $P(A|B)$ is the conditional probability of A given B.
- $P(B|A)$ is the condition probability of B given A.

Equation 2.1 is based on a strong *independence assumption*, which states that attributes $\mathtt{a}_1,\ldots,\mathtt{a}_n$ are conditionally independent of each other, given the value of the class label $\mathtt{C}$ [Friedman *et al.* 1997; Pandey and Pal 2011].

Naïve Bayes is considered to be an effective classifier, with several advantages compared to other classifiers. Its structure is given *a priori*, which makes it easy to construct, and it is also an efficient classifier [Cheng and Greiner 1999]. Another advantage is its ability to scan the training data only once, whilst using a small amount of the data for parameter estimation [Pandey and Pal 2011].

2.6.2 Clustering

A clustering technique divides data instances into groups or clusters, in which objects that belong to a cluster are more similar to each other, but different from objects belonging to other clusters [Sanderson 2008]. Good clusters are considered to be those in which the similarity between the members of a single cluster is great and in which the difference between clusters is also great. Clustering is an unsupervised learning process in which "a given collection of unlabelled patterns" is grouped "into meaningful clusters" [Jain *et al.* 1999]. According to Jain *et al.* [1999], the term *pattern* refers to "a single data item used by the clustering algorithm."

Clustering involves the following steps (as cited in Jain *et al.* [1999]):

1. Pattern representation (may include feature extraction and/or selection),
2. Defining an appropriate measure of proximity,
3. Clustering operation,
4. Data abstraction (optional), and
5. Assessing output (optional).

Some of these steps are illustrated in Figure 2.4 [Jain *et al.* 1999].

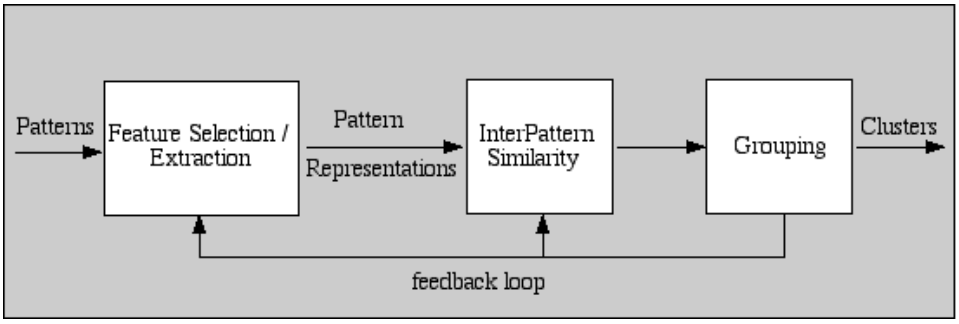


Figure 2.4: Clustering Steps

The different components shown in Figure 2.4 are described below:

- With *feature selection*, a subset of important features for creating clusters is chosen.
- *Feature extraction* involves the transformation of a larger input set of features into a smaller set of features.

- *Pattern representation* refers to the number of classes, patterns and features supplied to the clustering algorithm.
- *Inter pattern similarity* uses a distance measure to determine the similarity or dissimilarity between a pair of patterns.
- *Grouping* uses different techniques to produce clusters.

Clustering can be viewed as a form of classification. However, unlike classification, it does not require that the groups be known beforehand [Zaïane 1999]. Clustering techniques are used in different applications and sectors. Clustering also plays an important role in education. For example, Merceron and Yacef [2005] look at how clustering and cluster visualization can help in identifying particular behaviour amongst failing students. Another study by Jain and Gajbhiye [2012] compares three clustering algorithms: K-means, Farthest First and a hierarchical clustering algorithm. These algorithms are implemented in WEKA and the authors used them to produce models by applying them on the Iris, Haberman and Wine datasets. K-means outperformed the other two algorithms, producing the lowest percentage of incorrectly classified instances on all three data sets. In this research WEKA's Simple K-means clustering algorithm was used. K-means is considered to be one of the simplest and most popular clustering algorithms. Other clustering algorithms that were used in this research include Farthest First [Hochbaum and Shmoys 1985] and Expectation-Maximization (EM).

Simple K-means clustering

The K-means clustering algorithm uses a partition-based approach [Myatt and Johnson 2009]. The algorithm is initialized by selecting a value of k (the number of expected clusters). Thereafter k observations in the data are randomly selected as initial *centroids*. By calculating the Euclidean distance between the randomly selected centroids and the other observations in a dataset, the closest observations are grouped together with their related centroids. This is the initial clustering of the observations.

Equation 2.2 calculates the Euclidean distance (D_{ab}) between two points, a and b [Myatt and Johnson 2009]:

$$D_{ab} = \sum_{i=1}^{n} (\sqrt{a_i - b_i})^2 \tag{2.2}$$

After initial cluster formation, all data points are then reassigned to the new centroids. The above-mentioned process is performed recursively until no more data points are reassigned to clusters or it halts when an error (Err), expressed in Equation 2.3, occurs [Myatt and Johnson 2009].

$$Err = \sum_{i=1}^{k} \sum_{x=C_i} d_{x,\mu(C_i)} \tag{2.3}$$

where k is the clusters, x the observations and $\mu(C_i)$ the mean for the clusters.

The K-means algorithm is coded as follows [Revathi and Nalini 2013]:

K-means Algorithm
Input: I=$a_1, a_2, ...a_m$
k (number of clusters)
Output: K clusters
Procedure:
randomly assign initial k centroids
repeat
assign each a_i to a centroid using the closest distance
find new centroid
until no more reassigning or criteria met

Table 2.2: K-means Algorithm

Farthest First clustering

Farthest First implements the traversal algorithm developed by Hochbaum and Shmoys in 1985 [Witten and Frank 2005]. It is a variation of the K-means algorithm which, during the clustering process, places each centroid further away from the existing cluster centroids. According to Revathi and Nalini [2013], the Farthest First algorithm speeds up the clustering process and requires less reassignments and adjustments. Its time complexity is `O(nk)`, where `n` is the number of data instances and `k` the number of required clusters.

Expectation Maximization (EM) clustering

This clustering method is based on the Expectation Maximization algorithm. It begins by guessing the initial parameters, which it uses to calculate the cluster probabilities. The probabilities are used to recalculate the parameters [Witten and Frank 2005]. The process is repeated several times. The algorithm is divided into two steps: E-step for Expectation and M-step for Maximization. The code for this algorithm is shown in Table 2.3 [Mirkin 2012].

EM Algorithm
Start: select initial values for the parameters
E-step: given p_k, a_k (parameters) calculate g_{ik}
M-step: given g_{ik} calculate p_k, a_k (maximization of the log-likelihood function)
Halt: when the parameters do not change

Table 2.3: EM Algorithm

2.6.3 Association Rule Mining

Association Rule or Relationship Mining (ARM) is one of the data mining methods that were used in this research. Its purpose is "finding patterns, associations, correlations or causal structures among sets of items or objects in transaction databases, relational databases and other information repositories" [Chan *et al.* 2008]. In layman terms, ARM seeks to find patterns between items belonging in different sets and it produces *if-then* statements known as *association rules*. ARM is one of the most studied data mining algorithms [García *et al.* 2007]. It is commonly used and widely deployed in areas such as marketing where it is used, among other things, to detect regular patterns in consumer or customer behaviour [Abdullah *et al.* 2011; Chan *et al.* 2008]. Work in association rule mining has been extended to the educational domain, where the method is used in different ways to help provide answers to a variety of important and interesting questions. In order to significantly reduce the number of uninteresting rules

generated by the technique, some investigators prefer to use the *least association mining* approach that produces the least frequent but still interesting rules [Abdullah *et al.* 2011].

In educational research Association Rule Mining techniques are used in different ways. For example, they are used to identify patterns that characterize student performance, for extracting useful information to assist educators in making decisions about courses and for identifying patterns in learner behaviour [García *et al.* 2007]. In this research the Apriori algorithm found in the WEKA toolkit was used to mine association rules, using Computer Science I and first year performance datasets. The Apriori algorithm is the most commonly used association rule mining algorithm. Figure 2.5 summarizes the steps performed in this algorithm [Vannozzi *et al.* 2004].

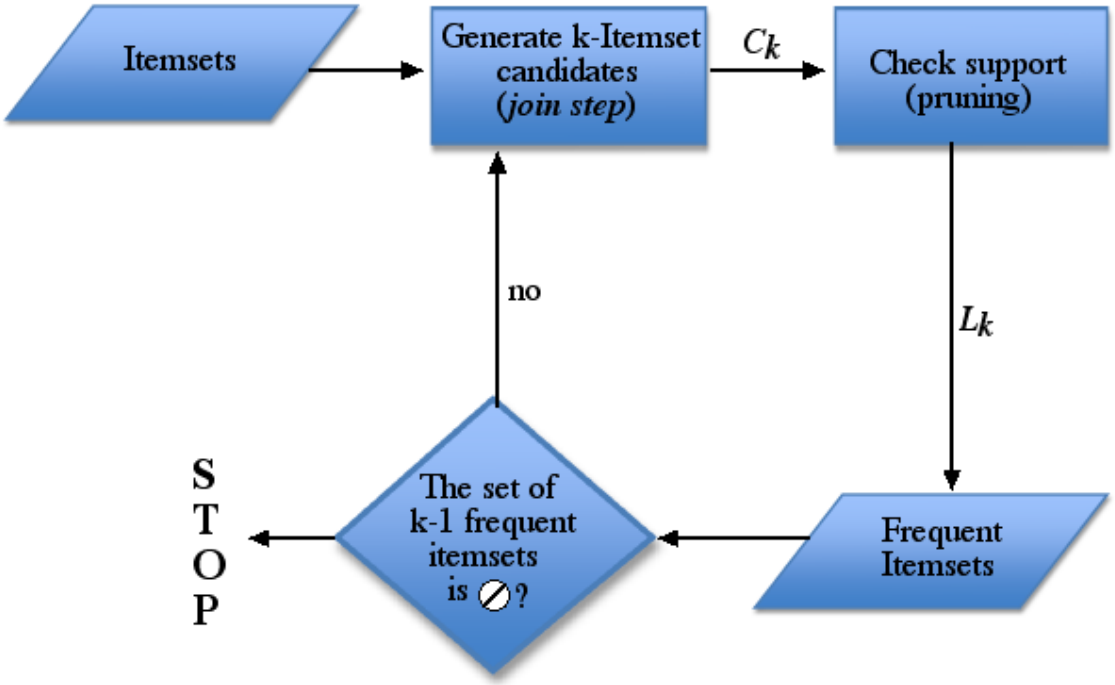


Figure 2.5: Apriori algorithm

Association rule mining has some drawbacks. Amongst others, it generates large volumes of rules or rules that are poorly understandable [García *et al.* 2007]. Furthermore, specifying the parameters for the algorithm requires some knowledge about it, which makes it less user friendly. This is one of the reasons the version in the WEKA toolkit was used. It requires a minimum number of parameters [García *et al.* 2007]. From WEKA the Predictive Apriori algorithm was selected and used. The algorithm returns a list of n best rules, where n is selected by the user [Sumithra and Paul 2010]. The *confidence* and *support* measures are combined to create the accuracy measure [Aher and Lobo 2012]. The value of the confidence measure indicates the strength of a particular rule, as well as its predictive ability, which is how likely the rule's consequent will occur, given the antecedents. The value of the support measure indicates the percentage of data instances that match a rule's antecedents. In their research Aher and Lobo [2012] incorporated and compared different association rule algorithms, including the Predictive Apriori algorithm, in a course recommender system.

2.6.4 Correlation and Regression Analysis

Correlation Analysis

Correlation analysis determines the relationship between two variables, based on two measures: *strength* and *direction* [Shi and Conrad 2009]. A *correlation coefficient* (r), whose value lies between -1 and 1, measures the strength of the relationship between two variables. When the value of r is -1, the relationship between two variables is a *perfect negative linear correlation*. When r is 1 the relationship is a *perfect positive linear correlation*. When r is 0, no correlation exists between two variables. The direction of the relationship between two variables may be either positive or negative, depending on whether the value of the correlation coefficient is positive or negative. The purpose of correlation analysis is to obtain a better understanding of the relationship between two variables. According to Taylor [1990], correlation analysis is the most widely used and reported statistical method for evaluating scientific research data. It is widely used across most fields of research. Correlation analysis assumes a linear relationship between two variables [Shi and Conrad 2009].

In correlation analysis different methods are used to determine the correlation coefficient. The most commonly used coefficient is the Pearson correlation coefficient [Taylor 1990]. The formula for calculating the Pearson correlation coefficient of a sample is shown in Equation 2.4 [Shi and Conrad 2009].

$$r_{xy} = \frac{\sum_i((x_i - \bar{x})(y_i - \bar{y}))}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}} \tag{2.4}$$

where $\bar{x}$ is the sample mean of variable x and $\bar{y}$ is the sample mean of the variable y and $i = 1, 2..., n$

According to Taylor [1990] the sign that precedes the correlation value does not indicate the direction of the relationship between two variables. Rather, it indicates that a negative correlation represents an indirectly proportional relationship while a positive correlation represents a directly proportional relationship between the variables. The strength of the relationship between two variables, using the correlation coefficient, has been defined in many ways. Table 2.4 shows how the different values of the correlation coefficient can be interpreted.

Table 2.4: Interpretation of Correlation Values [Dancey and Reidy 2004]

Correlation Coefficient (r)	Relationship Strength
0.7 - 0.9	Strong
0.4 - 0.6	Moderate
< 0.1 - 0.3	Weak
0	Zero

Regression Analysis

Regression Analysis investigates the relationship between a single dependent variable and one or more independent variables [Shi and Conrad 2009]. It depicts the relationship between a variable whose value is being predicted or estimated (dependent variable) and the variable or variables whose values are used to predict or estimate it (independent variables). Regression analysis, unlike correlation analysis, can be used to perform such tasks as modelling, prediction and estimation [Shi and Conrad 2009].

There are different types of regression techniques such as linear regression, non-linear regression and logistic regression. In this research the linear regression method was used to predict Computer Science I

scores using Mathematics I scores. In the linear regression technique a line equation is developed, with the best fitting line representing the data [Taylor 1990]. A simple linear regression is represented by the following equation [Shi and Conrad 2009]:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i \tag{2.5}$$

where:

- Y_i is the variable being predicted (dependent variable)
- β_0 is the intercept
- β_1 is the slope or gradient
- X_i is the variable being used to predict (independent variable)
- ε_i is the statistical error

The results obtained from the linear regression experiment that was performed in this research are presented in Section 4.5.1.

2.7 Conclusion

Several questions were posed in this research. For a better understanding of the work that needed to be done and to obtain clarity on the issues around these questions, a detailed review of the relevant literature was conducted. One of the most important aspects of this research is the application of data mining techniques on datasets collected in our educational context. Given the focus on data mining, this chapter began with a brief introduction and discussion of data mining, its goals, operations and techniques. This was followed by a discussion highlighting the use of data mining methods in different fields, such as commerce and science.

For a number of years data mining techniques have been used in different areas such as medicine, pharmaceutics, bioinformatics, life sciences, law, security, commerce and banking. Not until very recently has the educational sector, one of the major generators of data, been the focus of attention from a data mining perspective. Educational Data Mining (EDM) has emerged as a new research discipline which focusses, among other things, on the application and use of data mining techniques on educational contexts. A discussion of EDM was presented in Section 2.4, in which different examples of the research being conducted in this area were also given.

Given the nature of the questions that had to be answered in this research, purely statistical analysis methods could have been used to help answer some of these questions. Indeed, many studies have been conducted in educational and other types of research where these techniques been, and continue to be used. However, with EDM being a new research discipline, and with little or no EDM research being done in our School, faculty and university, a deliberate decision was then taken to conduct this research as an Educational Data Mining project. The EDM techniques that were selected and used to help answer the questions posed in this research were discussed in Section 2.5. The section also focussed in great detail on a discussion of the literature closely related to these questions. Some of the literature dealt with the first research question, which is about the relationship between race, gender and academic success, was discussed. The two factors, race and gender, are important in the South African context, given the country's history and their multifaceted role in our social, political, economic, educational and other contexts. Likewise, for the remaining research questions, a detailed study of the literature was conducted in order to obtain a better understanding of what needed to be done in order to address them.

Having decided to use data mining techniques in this research a decision had to be taken, given the nature of the questions that had to be answered, which data mining operations to use. Three operations were selected; namely classification, clustering and association rule mining. From each, different more methods were chosen. Three classification methods were chosen: J48, Decision Table and Naïve Bayes. Likewise, for the clustering operation, three clustering methods were selected: Simple K-means, Farthest First and Expectation Maximization. Only one association technique was used; namely the Apriori algorithm. Two statistical methods were also used: correlation analysis and regression analysis.

Chapter 3 presents and discusses the research method that was followed in conducting this research. Following the introduction, the chapter provides the motivation behind this study. Thereafter, the main research question and its related sub-questions are posed. A detailed methodology, including its associated phases, are presented and discussed, followed by a conclusion of the chapter.

Chapter 3

Research Methodology

3.1 Introduction

In Chapter 2 a detailed review of the literature was conducted which focussed, among other things, on data mining, data mining techniques and their application in educational environments in the form of Educational Data Mining (EDM). The chapter also laid a solid foundation and provided the background information necessary for developing a better understanding of the work that needed to be done in this research. This chapter presents and discusses the methodology that was followed in conducting the research. To a great extent the data mining aspect of the work relied on historical data about Computer Science I students, data which was collected between the years 2006 and 2012. Since previous data records were being used as part of the research it was decided that a longitudinal study, more specifically a retrospective longitudinal study, would be the most suitable methodological approach to follow.

Section 3.2 discusses some of the reasons for undertaking this research. These include the accumulation of a vast amount of unused data about Computer Science I students, the role played by different factors such as race, gender and course modules on performance in Computer Science I, the relationship between Mathematics I and Computer Science I, including the relationship between their respective modules, and so on. In Section 3.3 the methodology that was followed in conducting this research is discussed, including the CRISP-DM process model that was incorporated into it. The section begins, in Subsection 3.3.1, with a discussion on the longitudinal research paradigm. This is followed in Subsection 3.3.2 by a description of the research setting and the subjects involved in the study. One of the phases of the CRISP-DM process model, the *Data Understanding Phase*, involves the task of data collection. Subsection 3.3.4 includes a detailed discussion on data collection for this project. The section also includes a discussion on the software tools that were used for data modelling and analysis. A description is provided in Subsection 3.3.5 of the metrics and the statistical tests that were used in the study. The chapter is concluded in Section 3.4

3.2 Motivation for Research

A lot of data has over the years been collected by the School of Computer Science about its students. Some of this data is kept by other entities within the University such as the Faculty of Science, the Student Enrolment Centre and the Academic Information Systems Unit. The school has not done much, by way of research, to process and analyse this historical data in order to obtain meaningful information about the Computer science I students. Since opening its doors to students from all races Wits University, like other historically white universities, has become a higher education institution where diversity is the norm. Classrooms have likewise become diverse learning environments. Within the School of Computer Science diversity has become the norm. Classrooms are now filled with students from different racial,

gender, cultural and socio-economic backgrounds. For this research, it was felt necessary that this aspect of diversity in the Computer Science classroom be looked into, in order to study the relationship between factors such as race and gender, on the one hand, and performance in Computer Science I, on the other.

Currently, the Computer Science I course consists of four modules: BCO, DDS, FAC and LoC. Over the years, the performance of a significant number of Computer Science I students in LoC has been generally weak, if not poor. Apart from this fact being noted, nothing much has been done to look at the influence of LoC as a determinant of success or failure in Computer Science I. Given that a number of students have been failing this module, it was therefore deemed important to look into its effect on performance in Computer Science I.

Educational Data Mining (EDM) is a relatively new research discipline which focuses on information gathering and knowledge generation through the application of data mining techniques on educational datasets. Except for the research in EDM being conducted within the School of Computer Science, there is not much evidence available to indicate research activity in EDM by colleagues in other schools or faculties at Wits University. Furthermore, not much, by way of published work, points to EDM as being an area of research that other South African universities are engaged in. Given the increasing importance and adoption of data mining methods in educational contexts, this research looked at how, using selected data mining methods, first semester scores obtained in two Computer Science I modules, could be used to predict overall performance in Computer Science I at the end of the academic year.

Computer Science I students are required to take Mathematics I (Algebra I and Calculus I) as a co-requisite course. Over the years Mathematics I has constituted a major challenge for a significant number of Computer Science I students, resulting in some of them not being to able to proceed in Computer Science, given the fact that Mathematics is a pre-requisite subject for Computer Science II and Computer Science III courses. Despite the fact that Mathematics plays such an important role in determining the progress made by Computer Science students, systematic research has not been conducted within the School of Computer Science to study and analyse the relationship between Computer Science I and Mathematics I, and to determine the influence of Mathematics I on performance in Computer Science I. Obtaining such information could be very useful since this would, among other things, provide the school with a better understanding of some of the factors responsible for poor performance in Mathematics, and how improvements can be made to enable more students to succeed.

First-time first-year students often find it difficult to select the courses they would like to do. In most cases they rely on the guidance and advice of academics who register them to help them choose the courses they need to do. In some instances, they also rely on fellow students to help them choose their courses. Computer Science I students belong to this category. One aspect of this research was to determine which courses were closely associated with Computer Science I. This knowledge can help academics within the School of Computer Science to offer more objective, well-informed advice to first-time first-year Computer Science students, about which courses to select together with Computer Science I, thereby enhancing their chances of success in their first year of study. Another issue this research focused on was whether Computer Science I could be seen as a predictor in relation to the remaining group of courses selected by a first year Computer Science student. A question of interest was whether success or failure in Computer Science I could be used as an indicator of success or failure amongst the different courses selected by a Computer Science I student. If it is known that Computer Science I is a good predictor of performance, then first-time first year students could be encouraged to take it, to enhance their chances of success in first year, even though their intended major is not Computer Science.

3.3 Detailed Research Methodology

3.3.1 Research Paradigm - A Longitudinal Study

Universities collect vast amounts of data about their students. At Wits University data about students is collected and stored in several repositories, including those belonging to different faculties, schools and other entities, such as the Academic Information Systems Unit (AISU) and the Wits Academic Marks System (WAMS). The School of Computer Science has over the years accumulated and stored a lot of data about its students. For this research only first year student records were retrieved and used for analysis. Since previous academic records were being used, it was decided that a retrospective longitudinal study would be the most appropriate research approach. A longitudinal study is also referred to as an observational study, in which repeated observations of the same set of variables are made over a lengthy period of time [Menard 2002]. In this research the same variables were observed and analyzed over a 7-year period, between 2006 and 2012, inclusive.

There are different types of longitudinal studies. There are restrospective, cohort and panel longitudinal studies, to name a few [Menard 2002]. A retrospective longitudinal study looks at one or more specific characteristics of a population, back in time, in order to identify a trend. In this study previous records of first year Computer Science students were used for, among other things, for a similar purpose. A cohort study focusses, over a period of time, on specific groups (cohorts)[Menard 2002]. In such a study observations are made by collecting data from the individual members of a group, whose membership remains unchanged over time. A panel study involves the collection of data over time from the same set of people (sample or panel).

There are several instances where longitudinal studies have been conducted in educational and other environments to try and answer certain questions. The work of Howles [2007] highlights the fact that the general tendency amongst educators is to teach in a manner similar to how they were taught. However, given major transformations in educational technology and the fact that from year to year the students that are being taught differ in a number of respects, Howles [2007]'s view is that the traditional approach to teaching must be reviewed. Howles [2007] conducted a longitudinal study between 2004 and 2006 to address the following issues: what educators need to do in order to effect the changes required from them and how best to identify student needs in order to ensure their academic success. The study focussed on a number of factors (student values, learning styles and opinions) to determine whether they could be linked to academic performance, persistence and retention. One of the preliminary results from the study was that many students without "computer and programming language experience" may not have had opportunities before entering college, and that computer science educators assume that the students they teach will have had prior programming experience. It was established that there is a need for an introductory classes and adequate tutorial support for these students.

The longitudinal study by [Yurdugül and Askar 2013] compared male and female students in terms of their skills and knowledge of computer programming. Conceptual, syntactic and strategic aspects of knowledge were considered. The study used the latent growth model (LGM), which is a longitudinal statistical analysis method for estimating growth over time. There were significant differences in performance between males and females. The study by [Yurdugül and Askar 2013] used the latent growth model with longitudinal data while the current study uses the CRISP-DM process model was used with longitudinal data.

Other educational longitudinal studies include work by [White *et al.* 2009], which analysed the longitudinal performance scores of successful students in the online introductory programming course, the aim being to determine the effectiveness of teaching such a course online. In a study similar to that of [White *et al.* 2009], a management information systems course and a basic programming course were

offered to two types of students: online students and tradional students Ury [2005]. Compared to the traditional mode of delivery, the online mode of delivery was found to be effective. On the other hand, the average of the final performance scores obtained by traditional students was significantly higher than that obtained by their online counterparts. A comparative longitudinal study by [Mortagy and Boghikian-Whitby 2010] focussed on the issue of student beliefs about online education, and the longitudinal study by [Schmah *et al.* 2010] used data collected from patients recovering from strokes to compare 10 classification techniques. In this respect, [Schmah *et al.* 2010]'s study is similar to the current research, in which different classification techniques were also compared.

3.3.2 Research Setting and Subjects

Research Setting

This study was conducted in the School of Computer Science at Wits University. Of the 11 schools in the Faculty of Science, the School of Computer Science is the smallest, with the smallest number of academics and students. On average, between 150 and 200 students register annually to do the Computer Science I course (COMS1000). These students are drawn from different ethnic, social, economic and academic backgrounds. To be accepted to do Computer Science I at Wits University a student must satisfy the following entrance requirements, specified as part the National Senior Certificate (NSC) curriculum:

- The Admission Point Score (APS) must be at least 40 points. The APS of students who are accepted into the special Applied Computing (APC) curriculum stream must be at least 43 points.
- The English Home Language (EHL) and the English First Additional Language (EFAL) scores must be at least a 5 (60%-69%).
- The Mathematics score must be at least a 6 (70%-79%).
- Applicants whose APS scores lie in the range 38-39 are waitlisted. Those on the APC curriculum are waitlisted if their APS scores are in the range 38-42.

In a typical Computer Science I class female students are underrepresented and the majority of the students, particularly those coming from township and rural high schools, do not have prior programming experience. In order to try and address this and other educational shortcomings experienced by our Computer Science I students, a dedicated walk-in consultation venue has been made available, where students can go for consultations every day of the week. Consultation sessions with lecturers are also held. To give more time to students to work on their programming exercises and assignments, one of the first year laboratories is kept open for 24 hours every day, including week-ends. Tutorial sessions, for which senior students (honours, masters and doctoral) are responsible, are held for each of the four Computer Science I modules. In order to pass Computer Science I a student must obtain an average of at least 50% in the four modules, and the mark in each module must not be less than 40%, which is the sub-minimum mark.

Computer Science I students are able to select courses offered by different schools within the Faculty of Science. In addition to Computer Science I, these students must take Mathematics I as a co-requisite subject. Thereafter, they can select any two courses from a list that includes Computational and Applied Mathematics I, any of the courses offered in the biological sciences, Physics I, Chemistry I, Geography I, Geology I, Psychology I and so on. Table 3.1 contains a brief description of the four modules that are taught in Computer Science I [Sanders and Mueller 2000]:

When these modules were introduced by the School of Computer Science, the plan was to teach them in a specific sequence, over two semesters [Sanders and Mueller 2000]. In the first semester (Block 1 and

Table 3.1: Computer Science I Modules

Module Name	Module Description
Basic Computer Organization (BCO)	The content includes propositional logic, boolean algebra, link between logic & hardware, von Neumann model, etc.
Data & Data Structures (DDS)	Data representation, data structures, recursion, dynamic data structures, study of well known algorithms, verification, analysis, etc.
Fundamental Algorithmic Concepts (FAC)	Introduction to graph theory, proof techniques and simple proofs (direct, inductive, contradiction, constructive), formal specifications, mathematical model of a program, study of well-known algorithms, verify simple algorithms, analysis of simple algorithms, etc.
Limits of Computation (LOC)	Halting problem, responsibilities of scientists and professionals, ethics, the implications technology has for South Africa, the value of Computer Science, the value of research, overview of Artificial Intelligence and the Theory of Computation.

Block 2) BCO and FAC would be taught in parallel and in the second semester (Block 3 and Block 4), DDS and LOC would also be taught in parallel. Generally, the School of Computer Science has kept this arrangement. Occasionally, due to staffing constraints, this teaching sequence is not always adhered to.

Given our setting, and also the fact that the choice of different courses by individual Computer Science I students is bound to have an impact on their overall performance in Computer Science I and first year as a whole, some interesting questions arise, such as:

- How do the different course selections made by Computer Science I students contribute to their success or failure in first year.
- Which first year courses combine well with Computer Science I in terms of ensuring success in first year?
- How strong is the correlation between the performance scores in Mathematics I and those in Computer Science I?

Research Subjects

This research did not involve direct participation by human subjects. Instead, student records about Computer Science I students were used. These were collected between 2006 and 2012. Since the study did not involve direct human participation, it was decided to conduct it as an observational, longitudinal, retrospective study. Using the student records collected between 2006 and 2012, the study focussed on the following attributes (see Table 3.2):

3.3.3 Instrument: Success or Failure Determiner Tool

The main objective for creating a user interface, namely the Success Or Failure Determiner(SOFD) Tool, was two-fold:

1. To familiarize educational researchers within the university with a simple, easy-to-use software tool that can be used to learn more about the application of different data mining techniques in their research context.

Table 3.2: Attributes and Attribute Descriptions

Attribute	**Attribute Description**
BCOMark	Final mark for the BCO module
DDSMark	Final mark for the DDS module
FACMark	Final mark for the FAC module
LOCMark	Final mark for the LOC module
COMS1000	BCO, DDS, FAC and LOC marks combined
gender	female/male status of the student
race	African, White, Indian, Coloured or other
mathIMark	Mathematics I mark
thirdUnitMark	Mark obtained in third First-Year unit
fourthUnitMark	Mark obtained in fourth First-Year unit

2. To provide the opportunity to the Undergraduate Coordinator in the School of Computer Science to obtain information about first year students within the School, and to use such information for purposes such as performance prediction, identifying "at risk" students and preventing them from dropping out or failing by providing proactive assistance, learning about the individual and group characteristics of the students, and so on.

Several data mining algorithms implemented in the open-source WEKA toolkit were used. In this study WEKA was used as the back-end to the SOFD tool [Markov and Russell 2006], which provided the front-end interface. SOFD is intended to provide a platform for educational researchers and the Undergraduate Coordinator in the School of Computer Science, where they can be able to input student data and receive meaningful information as feedback from the system about those students.

Design

A high-level plan of the SOFD tool is presented in Figure 3.1. This is the simplest form of the design. It is clear from that level that the user will input data and the tool will output the Computer Science and First Year outcomes. Figure 3.2 shows a detailed view of the design, which includes the different components implemented in the tool. Although these plans show the high-level design of the user interface, the structure and navigation when using the tool is slightly different. The overall structure of the user interface is presented in Figure 3.3. The design is simple and the different parts of the tool are easy to navigate. Currently the tool is more suitable for use by data mining researchers, who understand the different data mining techniques incorporated in the tool, and who are therefore able to understand quickly how to use it and how to interpret its output. As part of future work, the tool can be extended to support academic and administrative members of staff within the School of Computer Science, who are always in close contact with students and, given sufficient training in using the tool, may use some of its results to provide informed advise to students.

Implementation

A complete snapshot of a single run through the SOFD tool is presented in Appendix G.1. The opening screen in the interface requires the user's access information, which enables her or him to continue using the tool. The opening screen is followed by a welcome page that introduces the tool and describes its features to the user, who is then required to choose whether they would like a Classification, Clustering or Association Rule Mining task to be performed. Each of these options is linked to the different factors being studied, as shown on Figure 3.3.

The WEKA toolkit forms the back-end of the SOFD tool, which was coded as the front-end, using Netbeans, Version 7.1. One of the uses of the SOFD GUI is to accept a list of student records such as,

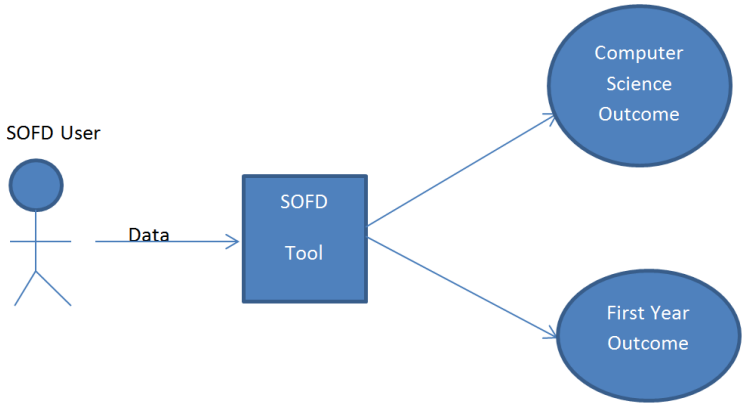


Figure 3.1: SOFD Tool High-Level Design

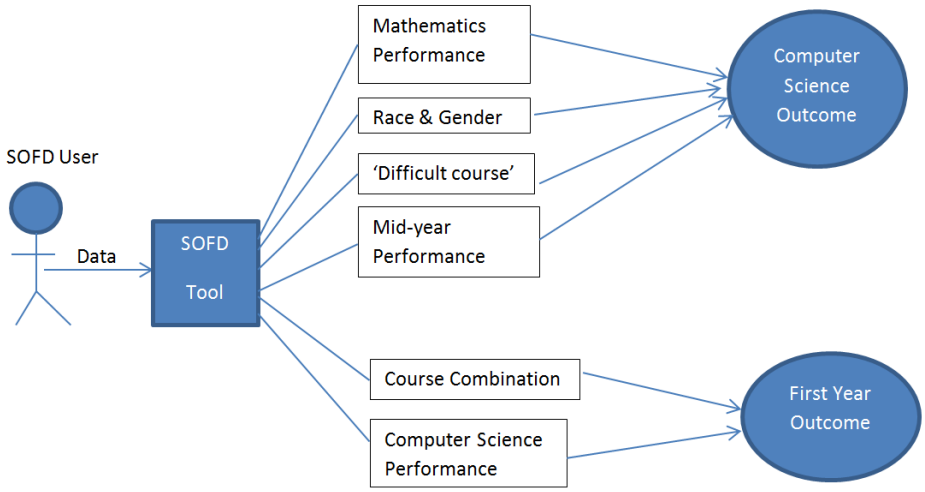


Figure 3.2: SOFD Tool Extended High-Level Design

for example, a list of the Mathematics I scores obtained by Computer Science I students. This is the test dataset. A predictive model, created during the training phase, is then applied on the test dataset. The result is displayed by showing the Mathematics I mark of each student alongside the predicted Computer Science I mark or symbolic grade, or alongside the Overall first year outcome of the student. The tool also displays useful statistics from WEKA.

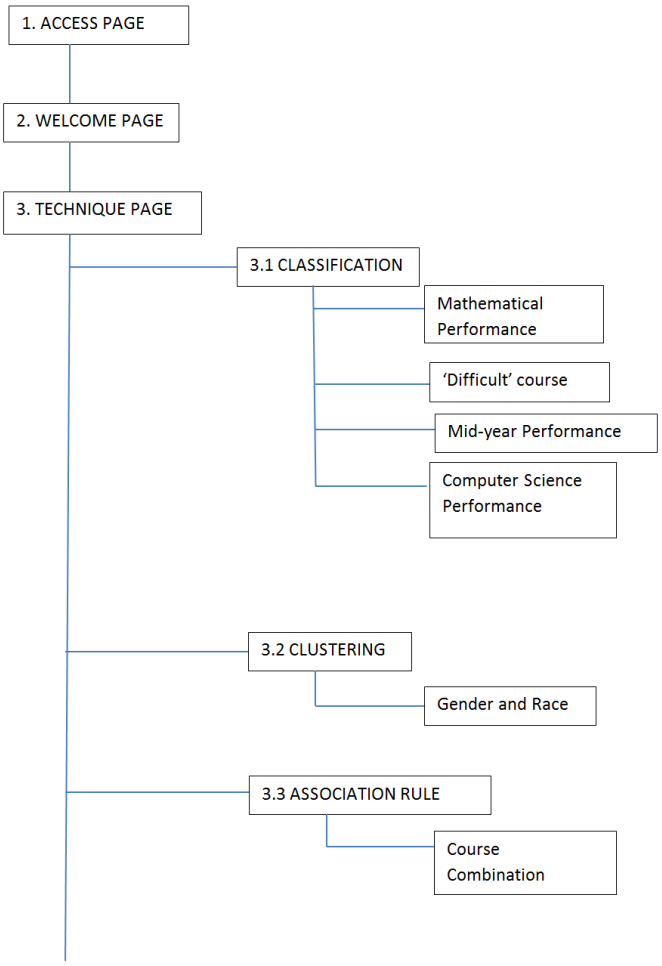


Figure 3.3: Structure and Navigation of the SOFD Tool

3.3.4 The CRISP-DM Process Model

Brief Introduction

Data mining played a significant and important role in this research. Even though the overall methodology was a retrospective longitudinal study, a data mining approach was incorporated as part of the overall methodology. Within the data mining community the CRISP-DM process model [Chapman *et al.* 2000] is increasingly being adopted as a standard by data mining practitioners working on a variety of data mining projects. The acronym CRISP-DM stands for Cross-Industry Standard Process for Data Mining. CRISP-DM was developed as a non-proprietary, freely available and application-neutral standard process for data mining projects [Larose 2014; Kabakchieva 2013]. It is one of a number of process-oriented frameworks that have been developed in the area of data mining. It describes the life cycle of a data mining project [Pechenizkiy *et al.* 2008]. CRISP-DM has also now been adopted by some educational researchers as a method of choice for doing research in the newly emerged discipline of Educational Data Mining (EDM) [Kabakchieva 2013; Kovacic 2010]. In CRISP-DM the life cycle of a data mining project consists of the six *phases*, which are illustrated in Figure 3.4: *Business Understanding Phase, Data Understanding Phase, Data Preparation Phase, Modeling Phase, Evaluation Phase* and *Deployment Phase*.

In this research the CRISP-DM process model was adopted as a guiding framework. The decision was taken in order to make it easier and convenient to undertake the project, by specifying and clarifying

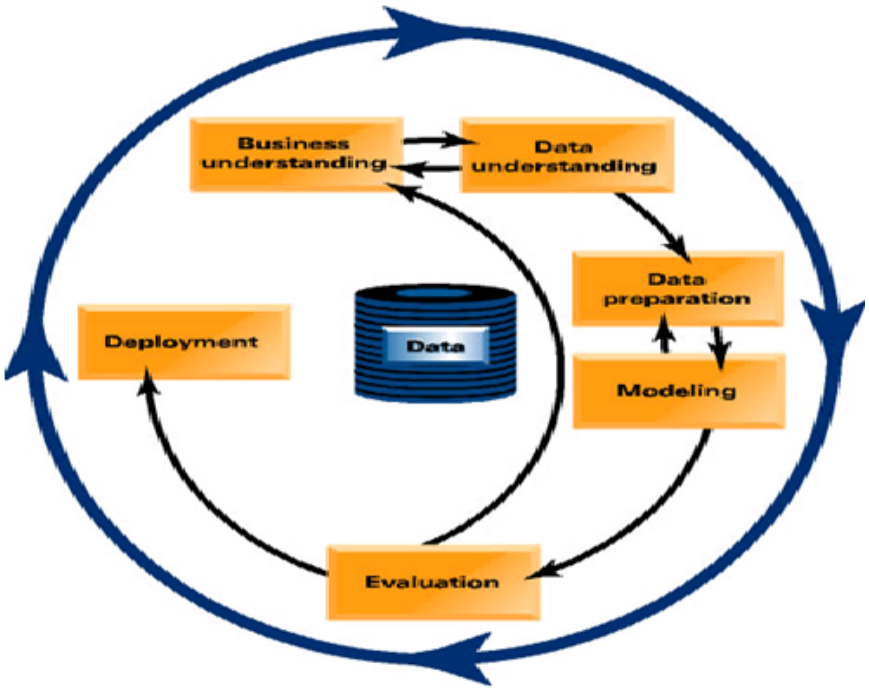


Figure 3.4: Cross-Industry Standard Process for Data Mining (CRISP-DM) [Builders 2012]

the tasks that needed to be done. This section describes and discusses, in relation to this research, the first four phases of the CRISP-DM methodology: *Business Understanding, Data Understanding, Data Preparation* and *Modeling.*

Business Understanding Phase

Throughout the world the drop-out and failure rates amongst first year students remain high [MacGregor 2007; Daley 2010; Bayer *et al.* 2012]. South African universities, including Wits University, are similarly affected by these phenomena, and the School of Computer Science is no exception. The first year students are a non-homogeneous and diverse group. They are drawn from different ethnic, gender, socio-economic and educational backgrounds. Given the teaching and learning environment this diversity, and more specifically the educational background of the students, presents a challenge that may hinder and negatively affect the success and smooth progression of a significant number of these students. An extensive study of the literature was conducted in order to develop a deeper and clearer understanding of how a variety of data mining techniques have been used to address some of the problems faced by first year students in situations similar to those in the School of Computer Science at Wits University.

The School of Computer Science is interested in developing a better understanding of the increasingly diverse group of Computer Science I students who attend its classes. Knowing these students better in terms of their diversity, strengths, weaknesses, backgrounds and other characteristics, will enable the school to determine how best to assist and empower them, and how to improve their chances of success in Computer Science I. Different classification methods were used to develop models for predicting the performance (failure or success) of these students. Other data mining techniques were also used, which included clustering and association rule mining techniques.

Data Understanding Phase

At Wits University different repositories are used to store data about students. Some of the data resides in individual schools and faculties and is also kept globally, at university level. When individual students apply or enroll to do a particular degree, the university's Student Information Management System (SIMS) is used to collect and store data about them. The data is stored in the University's Oracle database. The Wits University Academic Mark System (WAMS) is used to collect, store and process academic performance and other data about students.

Wits University has become a diverse educational institution, one that accepts students from different backgrounds, including race and gender. This diversity is a common feature in the first year classrooms in the university. The relationship between race and gender on the one hand and performance, on the other, is a widely debated researched topic. For instance, the role of gender in education is discussed in [Salazar *et al.* 2004; Siraj and Abdoulha 2009]. In this research demographic data, including race and gender, was collected about Computer Science I students, for the period 2006-2012. The final marks in Computer Science I were also collected. Each mark is assigned a Faculty decision code. Table 3.3 shows some of the decision codes used by the Faculty of Science to classify performance in particular courses.

Table 3.3: Decision Codes

Decision Code	**Description**
PAS	Pass
PMNP	Pass, may not proceed (used when a *fail* mark is condoned to a *pass*)
FAIL	Fail
PDS	Pass with distinction
FDEF	Fail, deferred examination refused
FNQL	Fail, did not qualify to write
FABS	Fail, absent from examination
WINC	Course/Topic Incomplete
FMNR	Fail, may not repeat course
WCSU	Suspended credit
PIN	Pass: Mark above allowed minimum
PINC	Pass - course incomplete
WAIT	Decision pending/result outstanding
WDEF	Deferred examination granted
WDEP	No mark from department
FSUB	Fail on sub minimum
FPAS	Pass, no credit

Computer Science I consists of four modules, which are taught concurrently with Mathematics I as a co-requisite unit. These modules, including their credits, are described in (Table 3.4).

Table 3.4: Computer Science I modules

Module	**Module Description**	**Points**
BCO	Basic Computer Organization	9
DDS	Data and Data Structures	9
LoC	Limits of Computation	9
FAC	Fundamental Algorithmic Concepts	9
MATH I	Mathematics I (Algebra I & Calculus I)	36

For this research data was obtained from the following entities:

- School of Computer Science Data Collection

  A total of 391 student records (Computer Science I), from 2010 to 2012, were obtained from the School of Computer Science. The data was used to address two questions. The first was to determine whether the LoC module, generally perceived to be difficult, could be considered as a predictor of performance in Computer Science I. Secondly, the mid-year average of BCO and DDS exam scores was used to determine whether the first semester performance in these two modules could be used to predict overall performance in Computer Science I at the end of the academic year. Of the 391 records a total of 257 records, collected in 2010 and 2011, were used as a training set. The remaining 134 records, collected in 2012, were used for testing purposes.

- Oracle Database Data Collection

  Some of the data required for this research was obtained from Wits University's Oracle Database. The data was retrieved using the Oracle Discoverer tool. For each Computer Science I student registered between 2006 and 2012, the following data items were collected: Computer Science I scores, Mathematics I scores and performance scores in two other first year courses. This data was used for three purposes. First, it was used to determine the relationship between Mathematics I and Computer Science I, and to determine the extent to which Mathematics I influenced Computer Science I. To answer this question, data collected during 2006 and 2012 on Computer Science I students was used. Another objective was to determine the strength of the association between the number of courses selected by individual Computer Science I students. For the period under study (2006-2011) Computer Science I students were registered for 39 different courses, excluding Mathematics I. These courses were offered by different schools within the Faculty of Science and, in some cases, by different schools in other faculties. A list of these courses is shown in Appendix J.1. Finally, there was also an interest to know whether the Computer Science I unit could be a performance predictor for the first year of study. For this part of the research final exam scores in Computer Science I, from 2006 to 2012, were used. To classify the overall outcome in a particular year of study, the Faculty of Science uses the overall result decision codes (Table 3.3.4).

Table 3.5: Overall Result Decision Codes

Decision Code	**Meaning**
PCD	Proceed
Q	Qualified
RET	return to year of study
CAN	cancelled
MBP	Minimum requirements not met. Renewal of registration permitted by Examinations Committee (Proceed)
MBR	Minimum requirements not met. Renewal of registration permitted by Examinations Committee (Return)
MRNM	Minimum requirements not met. Needs permission to reregister
XXXX	Result depends on outcome of deferred examination(s)
****	Result not available
FTC	Failed to qualify

Data Preparation Phase

Real-world data may be incomplete, noisy and inconsistent. It is therefore good practice to prepare the data by processing it before it used for data mining purposes. For this research first-year Computer Science records were obtained from the School of Computer Science and the University's Academic Information Systems Unit (AISU). In this phase the following tasks were carried out.

Data Selection

In order to be able to answer all the questions of interest in this research, we selected the the data attributes shown in Table 3.6.

Table 3.6: Selected Data Attributes

Attribute	**Description**	**Type**	**Values**
Race	Racial background	Discrete	⟨African,Chinese,Coloured,Indian,White⟩
Gender	Student's gender	Discrete	⟨Male,Female⟩
Year of Study (YOS)	Year One is considered	Discrete	⟨1⟩
Enrollment Year	Year studied	Discrete	⟨2006,2007,2008,2009,2010,2011,2012⟩
COMS1000 Mark	Computer Science I final mark	Continuous	⟨0.0-100.0⟩
BCO Mark	Final BCO mark (end of Semester I)	Continuous	⟨0.0-100.0⟩
DDS Mark	Final DDS mark (end of Semester I)	Continuous	⟨0.0-100.0⟩
FAC Mark	Final FAC mark (end of Semester II)	Continuous	⟨0.0-100.0⟩
LoC Mark	Final LoC mark (end of Semester II)	Continuous	⟨0.0-100.0⟩
Algebra Mark	Final Algebra module mark	Continuous	⟨0.0-100.0⟩
Calculus Mark	Final Calculus module mark	Continuous	⟨0.0-100.0⟩
Mathematics I	Mathematics I final mark	Continuous	⟨0.0-100.0⟩

Data Cleaning

Following from the main research question, several sub-questions had to be answered. For each of these the required data was collected, processed and analysed in order to provide the information necessary to answer each specific question. Table 3.7 shows the attributes and the data that was collected and used to answer the various questions.

- Part of the preparation for all the datasets was transforming the format of the file to the Attribute-Relation File Format (`.ARFF`). This is the standard format used in the WEKA tool kit. The file has a header which mentions the name of relations, columns and value types and a data section which consists of all the information (data records)[Kirkby 2008]. The training data set is the complete dataset while the test dataset has a "?" in place of the missing variable to be predicted. Examples of these two files are presented in the Appendix A.1.
- Some of the datasets for the individual research questions required more preparation. To look into the relationship between Mathematics I and Computer Science I, data from the Oracle database was extracted, after it had undergone the process of cleaning and integration. Since the dataset included both first time entering and returning first year students, some students were not enrolled for Mathematics I. These data records with missing Mathematics values were removed from the dataset. For Mathematics, the separate Algebra and Calculus marks were integrated into a single Mathematics I mark, as a weighted average of the marks obtained from the two modules.

Table 3.7: Research data sets

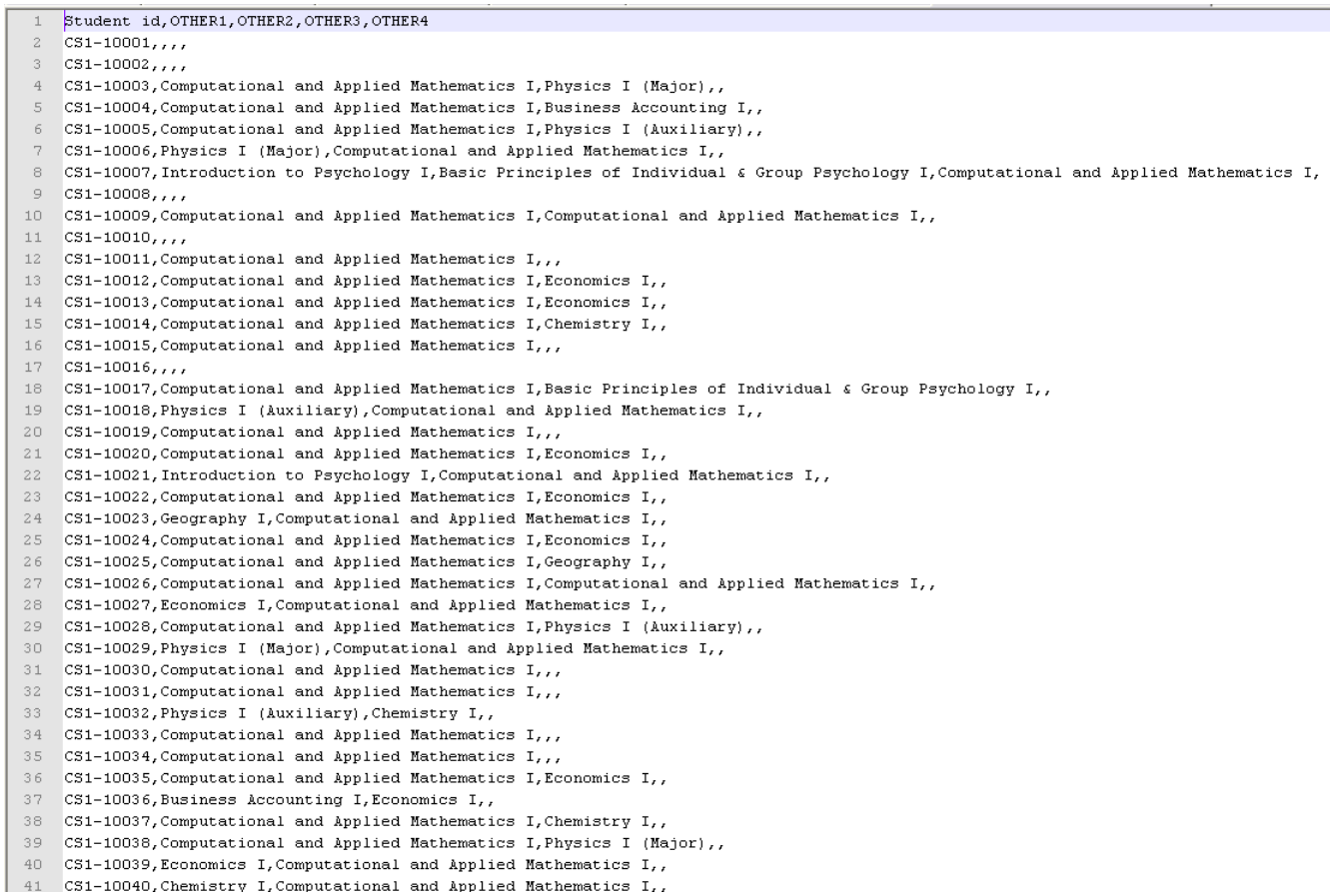
Question	Attributes	Records	Training Set	Test Set	Appendix
Race & Gender	race,gender,CS-1,school calendar year, year of study	**1337** (2006-2012)	**1118** (2006-2011)	**219** (2012)	
LoC Module	LoC,BCO,DDS,FAC,CS-1	**393** (2010-2012)	**259** (2010-2011)	**134** (2012)	C.1,C.2
Mid-year Performance	LoC,BCO,DDS,FAC,CS-1	**393** (2010-2012)	**259** (2010-2011)	**134** (2012)	D.1,D.2
Mathematics I & CS-1	Mathematics I, CS-1	**722** (2006-2012)	**609** (2006-2011)	**113** (2012)	A.1,A.2
Course Selection	19 First Year Courses	**564** (2006-2012)			E.1
CS-1 & Overall First Year	CS-1, Overall Result	**832** (2006-2012)	**698** (2006-2011)	**134** (2012)	E.1,F.1

- Additionally, first year students select different sets of courses to register into for the year. For this part of the research the aim was to determine the strength of the association between the set of courses selected by individual Computer Science I students. Depending on individual preference, and after taking the mandatory Computer Science I and Mathematics I courses, different courses from the same or different Faculties were selected by these students. From the data that was collected, only first year courses were considered. Also, courses with less than 10 Computer Science I students registered were excluded. This is similar to how it was done in a study carried out by Aher and Lobo [2012]. After further processing of the collected data, the number of course attributes was reduced from 39 to 19. Figure 3.5 shows some of the courses selected by Computer Science I students on registration.

Modelling Phase

Brief Review

Different data mining methods were used to obtain meaningful information from the data that was collected to carry out this research. These included prediction, classification, clustering and association rule mining techniques. *Prediction* can be used to determine what a future result will be. For example, the method can be used to predict by what percent the accident rate will have declined this time next year if the speed limit is reduced and constant on-the-road patrols are maintained by traffic officers. *Classification* determines the category or class to which a target variable belongs. For example, the technique can be used to determine whether someone applying for a bank loan is a low or high risk applicant. *Clustering* creates groups of data instances, in which data instances having similar characteristics are placed in their own, homogeneous group. For example, clustering can be used to group related emails or tweets into different clusters, depending on the topic of interest. *Association rule mining* is used to discover rules that describe the relationship between the attributes of a dataset. For example, the technique can be used to determine the proportion of alumni who respond positively to a call for donation to their university. Throughout this research the WEKA data mining toolkit was used as the back-end of the SOFD



```
Student id,OTHER1,OTHER2,OTHER3,OTHER4
CS1-10001,,,,
CS1-10002,,,,
CS1-10003,Computational and Applied Mathematics I,Physics I (Major),,
CS1-10004,Computational and Applied Mathematics I,Business Accounting I,,
CS1-10005,Computational and Applied Mathematics I,Physics I (Auxiliary),,
CS1-10006,Physics I (Major),Computational and Applied Mathematics I,,
CS1-10007,Introduction to Psychology I,Basic Principles of Individual & Group Psychology I,Computational and Applied Mathematics I,
CS1-10008,,,,
CS1-10009,Computational and Applied Mathematics I,Computational and Applied Mathematics I,,
CS1-10010,,,,
CS1-10011,Computational and Applied Mathematics I,,,
CS1-10012,Computational and Applied Mathematics I,Economics I,,
CS1-10013,Computational and Applied Mathematics I,Economics I,,
CS1-10014,Computational and Applied Mathematics I,Chemistry I,,
CS1-10015,Computational and Applied Mathematics I,,,
CS1-10016,,,,
CS1-10017,Computational and Applied Mathematics I,Basic Principles of Individual & Group Psychology I,,
CS1-10018,Physics I (Auxiliary),Computational and Applied Mathematics I,,
CS1-10019,Computational and Applied Mathematics I,,,
CS1-10020,Computational and Applied Mathematics I,Economics I,,
CS1-10021,Introduction to Psychology I,Computational and Applied Mathematics I,,
CS1-10022,Computational and Applied Mathematics I,Economics I,,
CS1-10023,Geography I,Computational and Applied Mathematics I,,
CS1-10024,Computational and Applied Mathematics I,Economics I,,
CS1-10025,Computational and Applied Mathematics I,Geography I,,
CS1-10026,Computational and Applied Mathematics I,Computational and Applied Mathematics I,,
CS1-10027,Economics I,Computational and Applied Mathematics I,,
CS1-10028,Computational and Applied Mathematics I,Physics I (Auxiliary),,
CS1-10029,Physics I (Major),Computational and Applied Mathematics I,,
CS1-10030,Computational and Applied Mathematics I,,,
CS1-10031,Computational and Applied Mathematics I,,,
CS1-10032,Physics I (Auxiliary),Chemistry I,,
CS1-10033,Computational and Applied Mathematics I,,,
CS1-10034,Computational and Applied Mathematics I,,,
CS1-10035,Computational and Applied Mathematics I,Economics I,,
CS1-10036,Business Accounting I,Economics I,,
CS1-10037,Computational and Applied Mathematics I,Chemistry I,,
CS1-10038,Computational and Applied Mathematics I,Physics I (Major),,
CS1-10039,Economics I,Computational and Applied Mathematics I,,
CS1-10040,Chemistry I,Computational and Applied Mathematics I,,
```

Figure 3.5: Courses selected by Computer Science I students

application that was specifically designed for use in this research.

Selected Modeling Techniques

Different data mining methods were selected to perform the tasks required in order to answer the questions that were posed in Chapter 1. Table 3.8 shows the algorithms that were selected to perform the different tasks. The cross validation method was used to evaluate the performance of the predictive models that were produced by applying the selected classifiers on the training datasets. The idea behind cross validation is to use a portion of the dataset, also known as the *validation dataset*, to test a model during training, thereby obtaining some idea from the results about the model's ability to handle unknown datasets. In this research the 10-fold cross validation technique, as implemented in WEKA, was used. Generally, k-fold cross validation, also known as *rotation estimation*, splits the dataset into k equal sets or *folds* [Kohavi 1995]. Of the k folds, (k-1) folds are used for training the model and the remaining fold (validation set) is used to test it (the value of k in our instance is 10).

- To perform the *clustering* task, the number of clusters was varied (k) in order to determine the most suitable cluster size. It is advisable, before settling on a particular cluster size, to experiment with different values of k until a "realistic" size has been determined [Jain *et al.* 2010]. The results for this part of the research are presented in Chapter 4.

- For *prediction* and *classification* the same three algorithms were used: J48, Decision Table and Naïve Bayes. They were applied on LoC, BCO, Mathematics I and DDS datasets that were used

Table 3.8: Selected Algorithms

Task	Algorithm	Purpose	Appendix
Race & Gender	Simple K-means, Farthest First Expectation-Maximization	Clustering	
LoC Module	J48, Decision Table and Naïve Bayes	Prediction	
Mid-year Performance	J48, Decision Table and Naïve Bayes	Prediction	H.2
Mathematics I & CS-1	J48, Decision Table and Naïve Bayes	Classification	H.1
Course Selection	Apriori	Association	H.3
CS-1 & Overall First Year	J48, Decision Table and Naïve Bayes	Classification	

to classify and/or predict performance in Computer Science I.

- Although the relationship between Mathematics and Computer Science was investigated using *classification*, regression and correlation analysis methods were also used.

- The *association rule mining* task was performed in two steps. First, a dataset consisting of 564 student records was used. There were 39 courses involved. In the second step the number of courses was reduced to 9, to focus only on those courses in which the number of Computer Science I students was highest. For this step, 528 student records were used. The Apriori association rule mining algorithm, implemented in WEKA, was used to generate a number of association rules, some of which produced relevant and meaningful information from the data.

3.3.5 Data Metrics Used

The following metrics (Equation 3.1 to Equation 3.5) were used to compare the classifiers selected for this study [Kumar and Rathee 2011]. The attributes in these metrics were derived from the confusion matrix which was created. A confusion matrix is a table normally used in supervised learning to depict the performance of an algorithm. Its columns represent the number of instances of a *predicted class*, and its rows represent the number of instances of an *actual class* [Visa *et al.* 2011]. The following abbreviations are used:

`TP`	**True Positive**
`TN`	**True Negative**
`FP`	**False Positive**
`FN`	**False Negative**
`FPR`	**False Positive Rate**

The true positive (TP), false positive (FP), true negative (TN) and false negative (FN) codes were identified as follows:

- TP: The number of instances that are correctly identified as passes (include PAS and PDS instances).

- FP: The number of instances that are incorrectly identified as passes (include FAL, FSB, FABS, WDF and WDA instances).

- TN: The number of instances that are correctly identified as deferred cases or failures (include WDEF, FAL instances).
- FN: The number of passes that are incorrectly identified as failures (include PAS,PDS instances).

Some of these decision codes are explained in Table 3.3.

The equations of the five metrics used are as follows:

- Sensitivity

  This is the proportion of correctly identified actual positive instances. The equation for sensitivity is as follows:

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.1}$$

- Specificity

  This is the proportion of actual negative instances that are correctly identified as such. The equation for this metric is as follows:

$$Specificity = \frac{TN}{TN + FP} \tag{3.2}$$

- Precision

  This is the proportion of positive instances that are correctly identified as such. The equation for this metric is as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3.3}$$

- Accuracy

  This is the proportion representing the total number of correct predictions Equation 3.4 defines this metric.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{3.4}$$

- False Positive Rate

  This is the proportion representing the total number of false positives. The equation for this metric is as follows:

$$FPR = \frac{FP}{FP + TN} \tag{3.5}$$

3.4 Conclusion

This research is motivated by the potential to extract knowledge about first year Computer Science students that could enhance their teaching and learning experience. This motivation is detailed in the beginning of this chapter. The chapter goes on to refer to the research methodology used. Since the research paradigm selected is a retrospective longitudinal study, a brief overview of longitudinal studies is done.

This is followed by the presentation of the research environment and the subjects. Additionally, the instruments used, namely the Success Or Failure Determiner tool and the CRISP-DM process were discussed. All the sub-questions were combined to create the SOFD tool. The modified version of the standard CRISP-DM methodology includes the collection and understanding of each data set and the pre-processing of the data. The modelling process included analysis of the data using different algorithms from the three selected techniques: classification, clustering and association rule mining. Finally, the data metrics that were used for analysis were presented.

Chapter 4 presents the results obtained from the work that needed to be done in order to answer the research questions that were posed, and thereby to obtain the answer to the main research question upon which this work was based.

Chapter 4

Results

4.1 Introduction

In Chapter 3 a detailed research methodology, based on the CRISP-DM process model, was presented and discussed. Several reasons were put forward as part of the motivation for this work. This chapter focuses on the work that was done and presents the results obtained in order to answer the questions that related to the main research question. This chapter is divided into six sections, each presenting the results attempting to answer a specific research sub-question. Section 4.2 presents the findings on the relationship between race and gender, on the one hand, and performance in Computer Science I, on the other. The following section, Section 4.3, then presents the results from the predicting of the final overall performance in the `COMS1000` unit using the "module perceived as being the most difficult". The third investigation in the research involved the inspection of the probability of using the first semester/mid-year mark to predict the final year-end Computer Science I mark. The results for this investigation are presented in Section 4.4. Thereafter results obtained when exploring the relationship between Mathematics I, Computer Science I and their modules are shown in Section 4.5. Section 4.6 then focuses on the effect of selecting certain course combinations with `COMS1000` and what impact that could have on overall first year academic performance. The final section goes a step further by considering the influence of academic performance in the `COMS1000` unit to overall first-year academic performance. The results of the findings in this investigation are presented in section 4.7. The chapter is then concluded with an overview of interesting and important findings.

4.2 Relationship between Race, Gender and Performance

For this research, the following question was investigated:

> *In a diverse Computer Science I classroom what is the relationship between race, gender and performance in the COMS1000 unit?*

A lot has been written in the literature about how factors such as race and gender may or may not be relevant as predictors of student academic performance [Salazar *et al.* 2004; Siraj and Abdoulha 2009]. In South Africa's social, economic and educational context, race and gender have in the past been used as key determinants of a person's status and destiny in society. In this research these two factors were used to look into their effect or influence, if any, on academic performance in Computer Science I.

In order to answer the question above a dataset consisting of 802 Computer Science I student records was used. These records were collected during the period 2006-2011. They were used as a *training set* for the clustering algorithms selected for this part of the study. For testing purposes a 2012 dataset of

Computer Science I records was used, with the same number of variables being used as in the training set. Three variables were included: `Race, Gender` and `COMSGrade`, the final mark obtained in the Computer Science I unit. Table 4.1 presents a descriptive summary of the training and test datasets.

Table 4.1: Descriptive Summary - Training and Test Datasets

Training Set (2006-2011)		Test Set 2012	
Variable	**Count**	**Variable**	**Count**
Race		**Race**	
Black	526	Black	104
White	142	White	25
Indian	110	Indian	12
Coloured	22	Coloured	1
Chinese	2	Chinese	0
Gender		**Gender**	
Female	180	Female	34
Male	622	Male	108
COMSGrade		**COMSGrade**	
A	79	A	13
B	55	B	9
C	158	C	31
D	197	D	46
F	313	F	43

Focusing on the training set, we note the following:

- The majority (65.5%) of students who studied Computer Science I during this period are Black (i.e. African in this instance).
- Generally, the Computer Science I classroom is racially diverse. However, Coloured and Chinese students are in the minority.
- Male students outnumber female students about 3.5 times, and 77.6% of the total number of students are male.
- About 61% (489) of these students passed Computer Science I, with about 10% (79) of the total number obtaining distinction passes ($\geq 75\%$) in Computer Science I.

Table 4.2 illustrates the correspondence between the symbolic grades and the actual scores assigned to them.

Table 4.2: Grades and Corresponding Scores

Grade	**Score (%)**
A	$\geq$75
B	70-74
C	60-69
D	50-59
F	40-49

Clustering is a descriptive, unsupervised data mining technique whose purpose is to place data instances that share similar characteristics into a group of their own. From a given dataset, clustering creates

data segments in which intra-segment similarities are maximized and inter-segment similarities are minimized. Selected clustering techniques were used, to look into the relationship or interaction between race, gender and performance in the Computer Science I unit. The following clustering algorithms were used: *Expectation Maximization (EM), Farthest First (FF), Make Density Based Clusterer (MDBC)* and *Simple KMeans (SKM)*. They were selected because of their availability in the WEKA toolkit, which was used as the back-end of the SOFD tool. Cluster analysis in WEKA allows for a data set to be modelled (put into clusters) and then another data set with a variable with an unknown value to be placed into the clusters from the pre-built model. In this research, this feature is used therefore there is a training data set which allows for comparison of the algorithms and building of models. It was decided that for each algorithm the number of clusters would be set to 5, to correspond to the five categories of grades (A,B,C,D and F).

4.2.1 Clustering - Using the Training Set

The clustering algorithms were applied on the training set consisting of 802 data records. Tables 4.3 to 4.5 summarize the results obtained after the FF, MDBC and SKM algorithms were applied on the training dataset. Each row in a table contains information about a particular cluster, including the number and the percentage of the instances it contains. For each cluster a single instance is identified as a centroid. This is an instance that characterizes or is typical of a particular cluster. For example, the centroid of cluster 0 in Table 4.3 is an instance for which the race of the student is "Black", whose gender is "Male" and who obtains symbol "C" in Computer Science I. For a centroid its attribute values are obtained by calculating the mean of the numerical attributes and determining the mode of the categorical attributes. Tables 4.3, 4.4 and 4.5 show the cluster performance results of the training dataset using three of the cluster algorithms, *FF, MBDC* and *SKM*, respectively. The results from the *EM* cluster analysis were different and are presented in Tables 4.6 and 4.7. Table 4.6 presents the percentage of the records which were placed in the five clusters. While Table 4.7 shows the distribution of race, gender and COMS Grade in the different clusters. The results from the *EM* run were loaded into the `Weka Clusterer Visualize` window.

Table 4.3: *FF* Cluster Performance Summary (Training Set)

Cluster no.	Clustered Instances	Percentage clustered	Race	Gender	COMS Grade
0	520	64.84%	Black	Male	C
1	101	12.59%	Indian	Female	D
2	98	12.21%	White	Male	D
3	66	8.23%	Black	Female	F
4	17	2.12%	White	Female	C

Table 4.4: *MDBC* Cluster Performance Summary (Training Set)

Cluster no.	Clustered Instances	Percentage clustered	Race	Gender	COMS Grade
0	322	40.15%	Black	Male	D
1	172	21.45%	White	Male	F
2	94	11.72%	Black	Female	F
3	190	23.69%	Black	Male	F
4	24	2.99%	Indian	Female	A

After running each algorithm on the same training set, the resulting model was saved in a file for later use during the testing phase. The following model files were created:

Table 4.5: *SKM* Cluster Performance Summary (Training Set)

Cluster no.	Clustered Instances	Percentage clustered	Race	Gender	COMS Grade
0	333	41.52%	Black	Male	D
1	153	19.08%	White	Male	F
2	95	11.85%	Black	Female	F
3	189	23.57%	Black	Male	F
4	32	3.99%	Indian	Female	A

- `EM-TRAINING.model`, created by applying the EM algorithm on the training set.
- `FF-TRAINING.model`, created by applying the FF algorithm on the training set.
- `MDBC-TRAINING.model`, created by applying the MDBC algorithm on the training set.
- `SKM-TRAINING.model`, created by applying the SKM algorithm on the training set.

Table 4.7 shows the output obtained from running the EM algorithm using the training set.

Table 4.6: *EM* Cluster Instances (Training Set)

Cluster no.	Clustered Instances (%)
0	280 (34.91%)
1	245 (30.55%)
2	108 (13.47%)
3	56 (6.98%)
4	113 (14.09%)

Table 4.7: *EM* Clustered Instances Distribution (Training Set)

	Cluster no.				
Attribute	0	1	2	3	4
Race					
Black	199.3853	200.6498	76.742	50.0979	4.125
White	15.5303	22.771	6.0786	5.3957	97.2244
Indian	47.5547	27.7169	19.2838	9.0998	11.3448
Coloured	9.0454	7.097	3.5483	2.2341	5.0754
Chinese	1.0612	1.981	1.0186	1.0063	1.9329
[total]	**272.5769**	**260.2157**	**106.6713**	**67.8337**	**119.7024**
Gender					
F	5.3968	6.1109	98.0701	59.5181	15.904
M	264.1801	251.1048	5.6012	5.3156	100.7984
[total]	**269.5769**	**257.2157**	**103.6713**	**64.8337**	**116.7024**
COMSGrade					
A	10.6743	4.3594	9.6062	3.7039	55.6562
B	24.7651	4.4921	1.9253	1.7417	27.0758
C	97.9898	5.8775	26.3734	4.5465	28.2128
D	133.2508	5.6758	6.0348	52.7558	4.2828
F	5.8969	239.8109	62.7315	5.0859	4.4748
[total]	**272.5769**	**260.2157**	**106.6713**	**67.8337**	**119.7024**

When dealing with numeric attributes the EM clustering technique uses the mean and standard deviation metrics to describe the clusters. In Table 4.7 three nominal (categorical) variables are used, namely race,

gender and COMSGrade. When using nominal attributes, clusters are described in terms of value counts, as shown in Table 4.7. To avoid zero probabilities, these values are incremented by 1, with a small value being used to adjust them. For example, looking at the race attribute, the 142 instances that represent white students are distributed across all the five clusters. In each cluster the number of instances is represented as a proportion of the total number of instances (142). Adding the different proportions produces the total number of 142 instances, incremented by some factor (in this case the factor is 5, which is the number of clusters used). Looking at Table 4.7 it can be seen that the majority of instances related to white students are to be found in cluster 4. This is a male-dominated cluster, with the highest number of As and Bs, and the second highest number of Cs. It has the lowest number of Ds and Fs. This is a cluster of high-performing students. For the other attributes (gender and COMSGrade), the EM technique uses a similar approach.

Race and Performance in Computer Science I

To determine the relationship between race and performance in Computer Science I, the `race` and `COMSGrade` attributes were plotted against each other, using the `Weka Clusterer Visualize` window. Figure 4.1 is a graphical depiction of the data, in which the relationship between race and performance is illustrated. This enables us to obtain a clear profile of these students. The cluster names are plotted on the X-axis and the symbolic, Computer Science I grades, on the Y-axis. Different colours have been used to represent the five student ethnic groups under consideration (i.e., Black,White,Indian,Coloured and Chinese).

Each cluster contains useful information about Computer Science I students, information we can use to draw a few conclusions about the nature of the relationship between race and performance.

- **Cluster 0**
  - This cluster contains a total of 280 (35%) students, the vast majority of whom are Black. There are some Indian and a few White students represented in this group. Most of the students in this group pass Computer Science I, the majority of them obtaining symbols `C` and `D`, the latter being the symbol that best characterizes students who are members of this cluster. This is the group in which Black students obtain the highest number of `A`s, and the number of failures (grade F) is also very low.

- **Cluster 1**
  - There are 245 (31%) students in this group, which is characterized by a very high failure rate, especially amongst Black students, who constitute the overwhelming majority of members. There are only a few Coloured, Indian and White students who belong to this poor-performing group. In academic terms, this group of students can be viewed as a "Problem Group". Their members risk expulsion or exclusion on academic grounds, may require academic counselling, are most likely to be "at-risk" students, and may need to be placed in a academic retention programme.

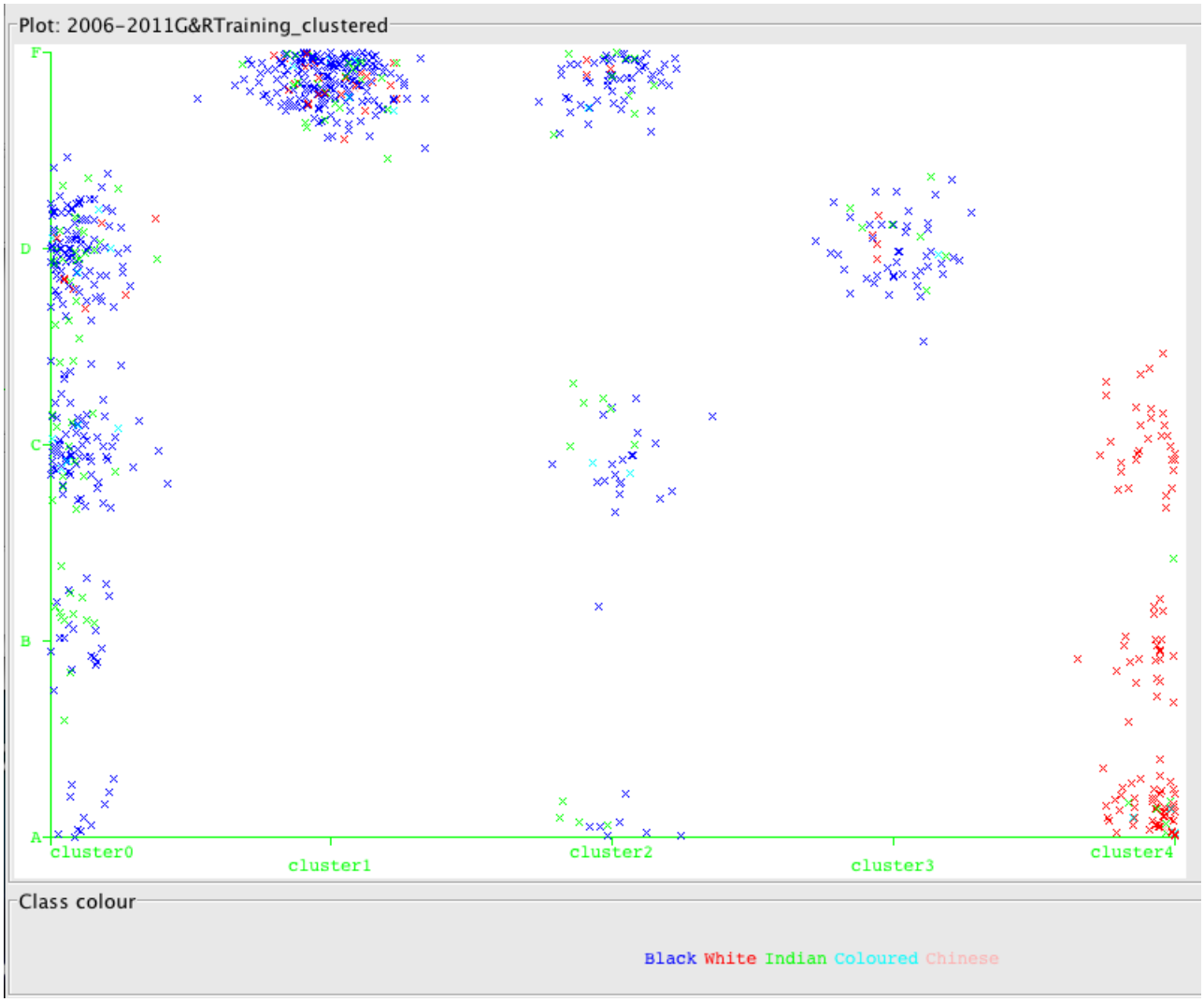


Figure 4.1: Cluster Visualization for Race and Performance in Computer Science I using training data set

- **Cluster 2**

  - This group consists 108 (13%) members. It is mostly dominated by Black students. Even though the number of failures in this group is not as high compared to *Cluster 1*, the failure rate is nonetheless still high, especially amongst Black students. A few Indian and White students also belong to this group of failing students. Of the number of students who pass Computer Science I, the majority of them obtain symbol C. This group consists of Black and Indian students, who also make up the smaller number of students who obtain As in Computer Science I.

- **Cluster 3**

  - There are 56 (7%) students in this group of "Fifty-Something Percenters". This is a group consisting mostly of Black students, with a few Coloured, Indian and White students amongst them. In terms of their academic performance, the vast majority of these students obtain Ds (50-59%) in Computer Science I.

- **Cluster 4**

  - There are 113 (14%) students represented in this cluster. This is a group of "High Performers", which is almost totally dominated by White students. Of all the five groups, this is the group with the highest number of `A`s, and a high number of `B`s and `C`s. Generally, this is a good group of Computer Science I students.

Gender and Performance in Computer Science I

There are ongoing studies that look into gender and its relationship to performance in different subjects. This research has taken a similar approach, by looking into the relationship between two attributes, `gender` and `COMSGrade`, that were used in the clustering experiments. Figure 4.2 shows the clusters that depict the relationship between gender and performance in Computer Science I.

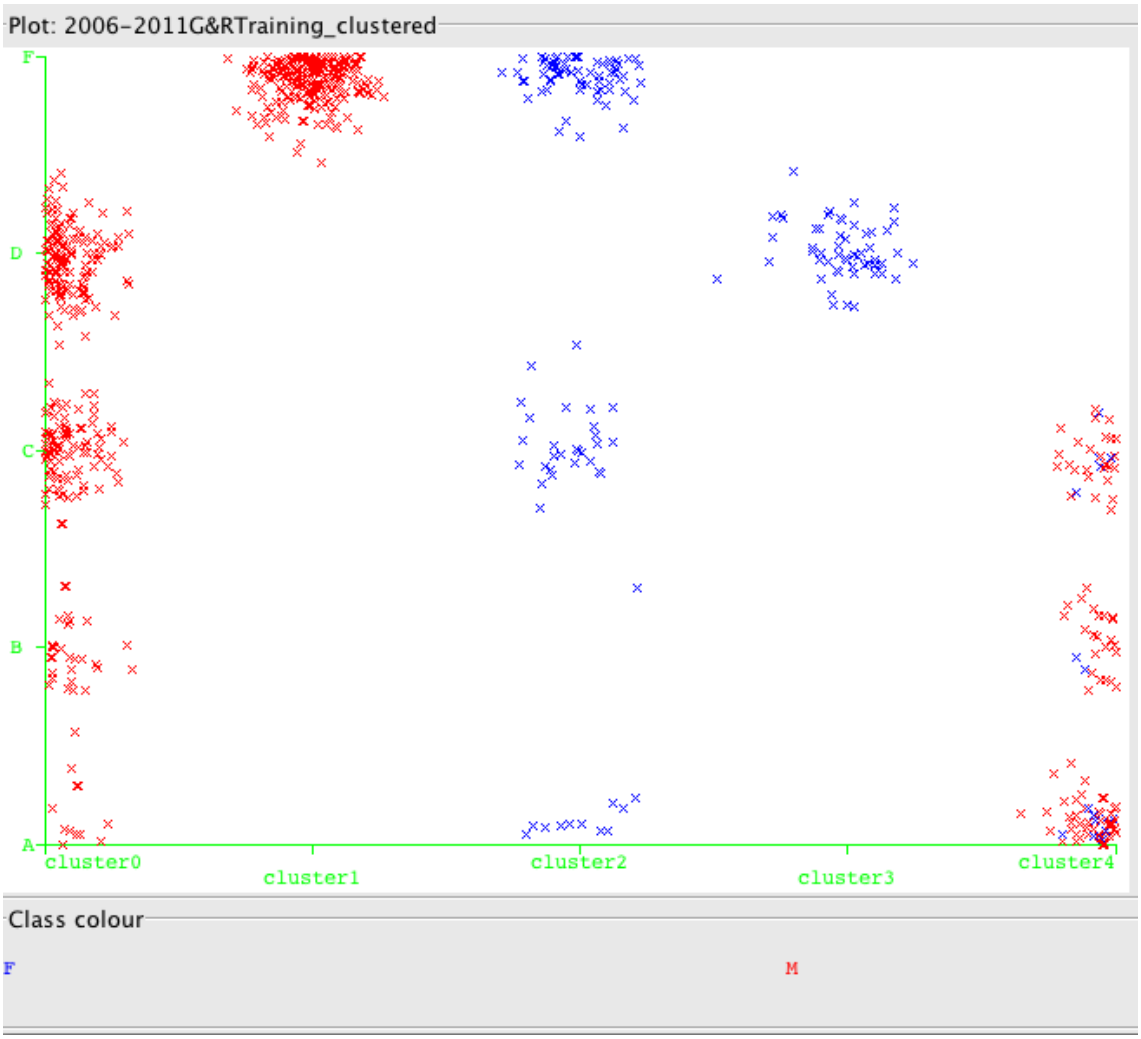


Figure 4.2: Gender and Performance in Computer Science I (training dataset)

By looking at the clusters in Figure 4.2, a few conclusions can be drawn regarding the relationship between gender and performance in Computer Science I.

- **Cluster 0**
  - This is an *all-male* cluster whose members are doing well in Computer Science I, with the majority of students obtaining `C`s and `D`s. Compared to these, a fewer number of students are obtaining `A`s and `B`s. There are virtually no failures (grade `F`) in this group.
- **Cluster 1**
  - This is another *all-male* cluster. However, unlike Cluster 0, where all the students are succeeding academically, the students in this cluster are all failing, obtaining symbol `F`.
- **Cluster 2**
  - This is an *all-female* cluster, in which the failure rate is quite high, although not as much as the failure rate amongst male students in Cluster 0. As in Cluster 0, a sizeable number of students in this group obtain `C`s in Computer Science I, with a smaller number obtaining `A`s.
- **Cluster 3**
  - All the students in this *all-female* cluster are successful, obtaining `C`s in Computer Science I.
- **Cluster 4**
  - This is the only mixed-gender group, in which male students far outnumber female students. The members of this group are good performers, obtaining more `A`s compared to students in Cluster 0 or Cluster 2. The lowest symbol in this group is a `C` (60-69%).

Table 4.8: Results of the cluster analysis on the test data set

Cluster Number	**Cluster Model**			
	EM (`EM.model`)	FF (`FF.model`)	MDBC (`MDBC.model`)	SKM (`SKM.model`)
0	57 (40%)	96 (68%)	72 (51%)	70 (49%)
1	35 (25%)	19 (13%)	24 (17%)	22 (15%)
2	18 (13%)	16 (11%)	13 (9%)	15 (11%)
3	14 (10%)	8 (6%)	29 (20%)	29 (20%)
4	18 (13%)	3 (2%)	3 (4%)	6 (4%)

4.2.2 Clustering - Using the Test Set

The training phase produced different models by applying selected clustering algorithms on the training set. Each of the models was then tested to see how well it would perform in clustering instances from a given test set. The different models were applied on the test data set, producing the following clusters (Table 4.8). There were 142 records in the test data set, which consisted of the final marks obtained in Computer Science I in 2012. Again, the results obtained from the EM model's output were used to visualize the cluster assignments in the `Weka Clusterer Visualize` window.

Race and Performance in Computer Science I

The different models produced during the training phase were used on the 2012 test dataset, which included the final Computer Science I marks for that year. The following clusters (Figure 4.3) were produced by applying the EM model on the test dataset.

The clustering pattern is similar to that produced by the EM classifier on the training set (see Figure 4.1). *Cluster 0*, the largest cluster, consists mainly of Black students, and very few Indian and White students. There are no failures in this group. The bulk of these students, mainly Black, obtain `C`s and `D`s in Computer Science I, and very few of them obtain `A`s and `B`s. *Cluster 1*, the second largest cluster, represents mainly Black, some White and very few Indian and Coloured students. This group may be labelled as *Underperformers*, since all of these students have failed Computer Science I (grade `F`). The profile of students in *Cluster 2* is somewhat similar to that of *Cluster 0*, the only difference being the absence of `A`s and `D`s in *Cluster 2*. The group of students in *Cluster3* can best be described as *Fifty Percenters*. The mainly Black students in this group all pass Computer Science I, obtaining between 50 and 59 percent (grade `D`). *Cluster 4* represents mainly White students, who get more `A`s than any other race group. This is a *High Performers* cluster. There are no `D`s and `F`s in this group, although there are White students in *Cluster 1* who have not passed *Computer Science I*.

From these results, the following conclusions can be drawn regarding the relationship between race and performance in Computer Science I:

- The majority of Black students who pass Computer Science I obtain `D`s (see *Cluster 0* and *Cluster 3* in Figure 4.3). A smaller number of these students also obtain `C`s (see *Cluster 0* and *Cluster 2*).
- The failure rate amongst Black students is much higher compared to failure rates amongst students of other races (see *Cluster 1* and to a lesser extent, *Cluster 2*).
- White students obtain more `A`s (75-100%) compared to students from other race groups. Their failure rate is also lower, compared to that of Black students.

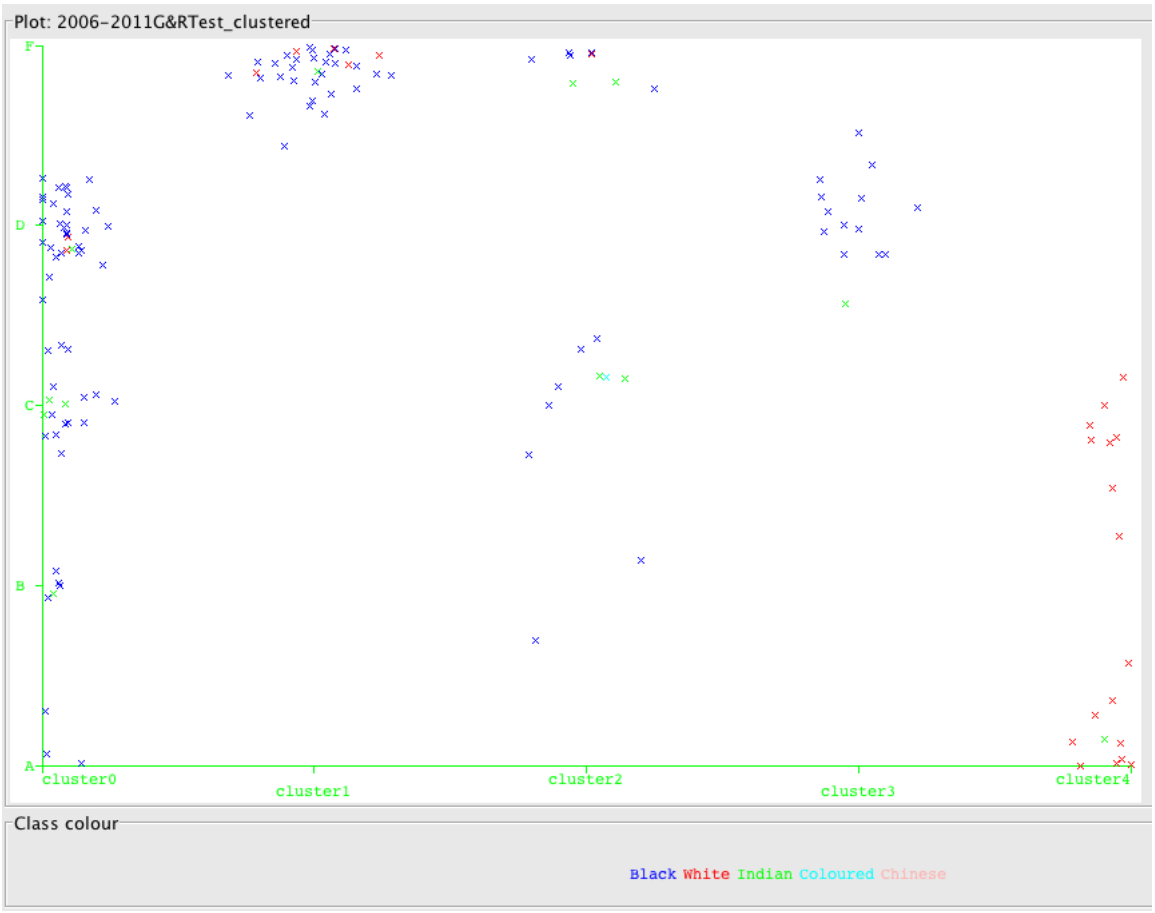


Figure 4.3: Cluster Visualization for Race and Performance in Computer Science I using Test data set

From Figure 4.3 it is visible that majority of the blue crosses (black students) are in the C, D and F grading regions. While majority of the red crosses (white students) are in the A and C regions. There are many reasons that could influence this. These include but are not limited to the language of instruction being a barrier for black students. English is not a home language for most black students and for this reason some black students struggle with understanding English and its concepts. Additionally other factors such as cost of tertiary education, transportation and resources also affect a lot of black students state of mind and therefore retention [Steyn *et al.* 2014]. Previous experience and a foundation of working with computers is also important when studying a course like Computer Science. Hoffman and Novak [1998] discussed a majority of white people having access to computers at home (42% compared to 29%) and having used the web (26% compared to 22%). Although these are statistics in the States, literature in South Africa has shown around 13.6% of South African households having access to computers in 2003 [Tlabela *et al.* 2007].This will have obviously increased in recent years. However with blacks being in the majority in the country, it is most likely that there will be a higher number of black households with no access to computers.

Due to space constraints, not all the results produced by the different models on the test data can be presented for visualization, as in Figure 4.3. It was decided to summarize the results and present them in the form of a table, using a coding scheme to represent, for a given cluster, the presence (`Y`) or absence (`N`) of a particular race group in a given grade. For example, if we consider *Cluster 0* and the EM model, we see that under column `A` of Table 4.9, the code `Y N N N N` is used. It means that in *Cluster 0*, only Black students have obtained symbol `A`, and not other race groups (that is, White (`W`), Indian (`I`), Coloured (`C`) and Chinese (`Ch`)) Looking at *Cluster 1* and column `F`, we notice that using the MDBC model a number of White students have failed, including a *smaller* number of Indian students (use of

lowercase y). There are no failures amongst Black, Coloured and Chinese students.

Table 4.9: Cluster Summary Results - Race and Performance

Cluster	Model	A Race					B Race					C Race					D Race					F Race				
		B	**W**	**I**	**C**	**Ch**	**B**	**W**	**I**	**C**	**Ch**	**B**	**W**	**I**	**C**	**Ch**	**B**	**W**	**I**	**C**	**Ch**	**B**	**W**	**I**	**C**	**Ch**
0	EM	Y	N	N	N	N	y	N	y	N	N	Y	N	y	N	N	Y	y	y	N	N	N	N	N	N	N
	FF	Y	N	y	N	N	Y	N	y	N	N	Y	Y	Y	N	N	Y	N	N	N	N	Y	N	y	N	N
	MDBC	Y	N	N	N	N	Y	N	N	N	N	Y	N	y	N	N	Y	y	y	N	N	N	N	N	N	N
	SKM	y	N	N	N	N	Y	N	y	N	N	Y	N	y	N	N	Y	y	y	N	N	N	N	N	N	N
1	EM	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	Y	y	N	N
	FF	N	N	N	N	N	N	N	N	N	N	N	N	y	N	N	Y	N	y	N	N	N	N	y	N	N
	MDBC	N	Y	y	N	N	N	y	y	N	N	N	Y	N	N	N	N	N	N	N	N	N	Y	y	N	N
	SKM	N	Y	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	y	N	N
2	EM	N	N	N	N	N	y	N	N	N	N	Y	N	y	N	N	N	N	N	N	N	y	y	y	N	N
	FF	N	Y	N	N	N	N	y	N	N	N	N	N	N	N	N	N	y	N	N	N	N	Y	N	N	N
	MDBC	N	N	N	N	N	N	N	N	N	N	Y	N	y	N	N	N	N	N	N	N	Y	N	y	N	N
	SKM	N	N	N	N	N	y	N	N	N	N	Y	N	y	N	N	N	N	N	N	N	Y	N	y	N	N
3	EM	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	y	N	N	N	N	N	N	N
	FF	N	N	N	N	N	y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	y	N	N	N
	MDBC	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N
	SKM	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N
4	EM	N	Y	y	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
	FF	N	y	N	N	N	N	N	N	N	N	N	N	y	N	N	N	N	N	N	N	N	N	N	N	N
	MDBC	N	y	N	N	N	N	N	N	N	N	N	N	y	N	N	N	N	N	N	N	N	N	N	N	N
	SKM	N	y	y	N	N	N	N	N	N	N	N	N	y	N	N	N	N	N	N	N	N	N	y	N	N

Table 4.10 groups the algorithms together, in terms of their similarity in assigning performance scores to different clusters.

Looking at Table 4.10, we observe the following:

- At least three (EM,MDBC,SKM) of the algorithms identify *Cluster 0, Cluster 2* and *Cluster 3* as grade A clusters.
- At least three ({EM,MDBC,SKM} and {EM,FF,SKM}) of the algorithms identify *Cluster 0, Cluster 3* and *Cluster 4* as grade B clusters.
- At least three ({EM,MDBC,SKM} and {FF,MDBC,SKM}) of the algorithms identify *Cluster 0, Cluster 2, Cluster 3* and *Cluster 4* as grade B clusters.
- At least three ({EM,MDBC,SKM} and {FF,MDBC,SKM}) of the algorithms identify *Cluster 0, Cluster 2, Cluster 3* and *Cluster 4* as grade C clusters.
- At least three ({EM,MDBC,SKM} and {FF,MDBC,SKM}) of the algorithms identify *Cluster 0, Cluster 1, Cluster 2, Cluster 3* and *Cluster 4* as grade D clusters.
- At least two ({MDBC,SKM}) of the algorithms identify *Cluster 0, Cluster 1, Cluster 2* and *Cluster 3* as grade F clusters.

Table 4.10: EM,MDBC,SKM and FF Clustering Results - Race and Performance
Computer Science I Grade

Cluster	A	B	C	D	F
0	*(EM,MDBC,SKM)* FF (close)	(EM,FF,SKM) MDBC (close)	*(EM,MDBC,SKM)* FF (close)	*(EM,MDBC,SKM)* FF	*(EM,MDBC,SKM)* FF
1	(EM,FF) (**MDBC,SKM**)	(EM,FF) MDBC SKM	(EM,SKM) FF MDBC	*(EM,MDBC,SKM)* FF	(**MDBC,SKM**) FF (close) EM
2	*(EM,MDBC,SKM)* FF (close)	(EM,SKM) FF MDBC	*(EM,MDBC,SKM)* FF	*(EM,MDBC,SKM)* FF (close)	(**MDBC,SKM**) EM (close) FF
3	**All**	*(EM,MDBC,SKM)* FF (close)	**All**	(FF,MDBC,SKM) EM	(**MDBC,SKM**) FF (close)
4	(EM,SKM) (FF,MDBC)	**All**	(FF,MDBC,SKM) EM	**All**	(EM,FF,MDBC) SKM (close)

It can be seen from this information that the selected clustering algorithms show strong agreement in most cases in their assignment of Computer Science I grades to their respective clusters.

Gender and Performance in Computer Science I

This research also looked at the relationship between gender and performance in Computer Science I. Figure 4.4 depicts this relationship, following the EM model's execution using the test dataset.

Looking at Figure 4.4 and with the exception of *Cluster 4*, there is not much of a "gender mix" within individual clusters. For example, *Cluster 0* and *Cluster 1* are all-male clusters, whereas *Cluster 2* and *Cluster 3* are all-female clusters. Amongst the students who obtain As in *Cluster 4*, there are male and female students represented. This cluster also shows male students to be doing quite well. There are no failures and the lowest symbol is a C. Similarly, male students are doing well in *Cluster 0*. There are a few As (less than in *Cluster 4*) and no failures. As can be seen in *Cluster 1* and *Cluster 2* respectively, more male students are failing Computer Science I compared to female students. These findings are summarized as follows:

- A greater percentage of male students obtain As, compared to female students. The majority of these male students are White.
- More males are failing Computer Science I compared to females. The majority of male students failing Computer Science I are Black.

,
Table 4.11 summarizes the results obtained using the models whose results, due to space constraints, cannot be visually represented.

Table 4.12 groups the algorithms together, in terms of their similarity in assigning performance scores to different clusters.

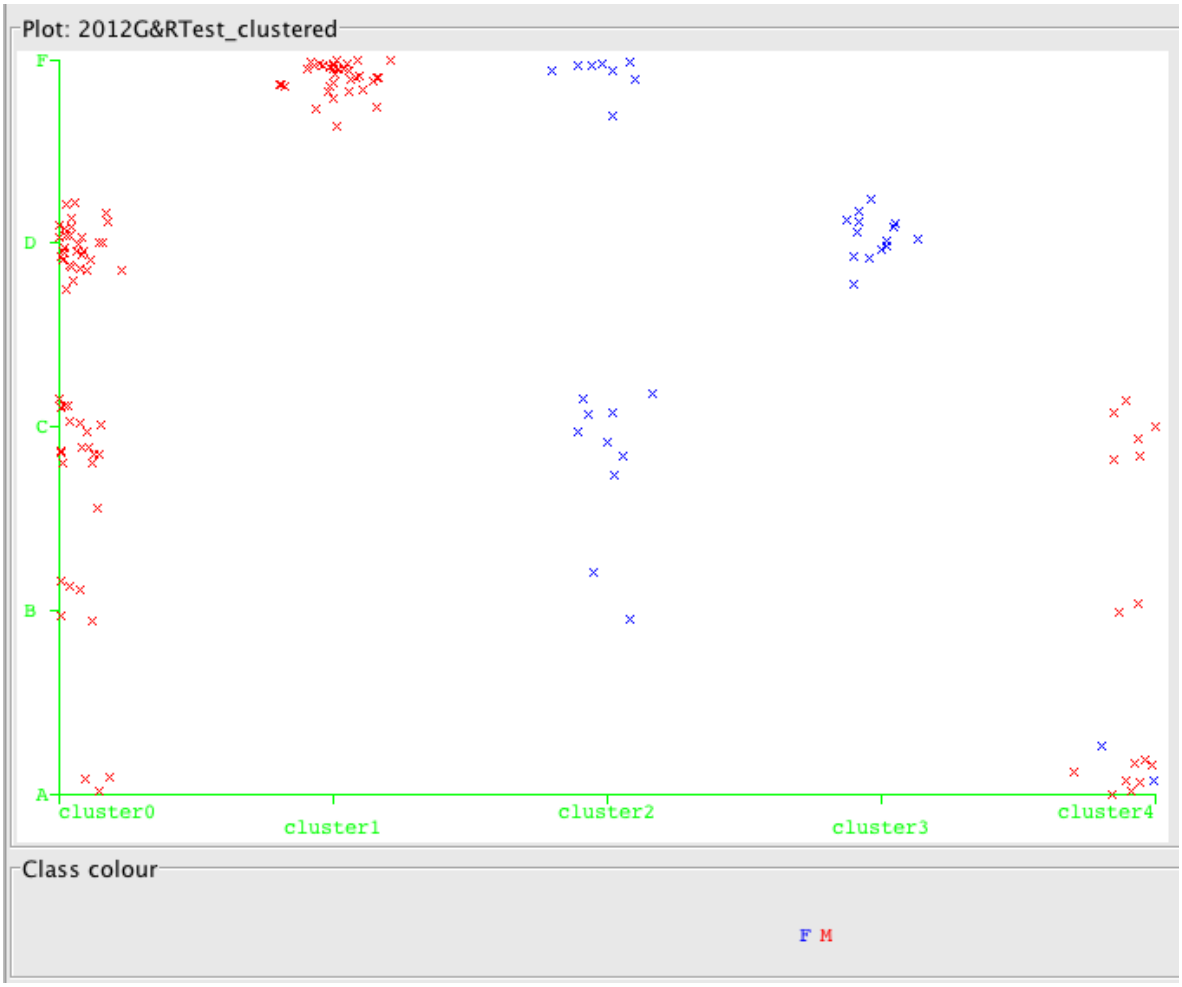


Figure 4.4: Cluster Visualization for Gender and Performance in Computer Science I using Test data set

Looking at the contents of this table a similar conclusion can be drawn, as for Table 4.10, that the clustering algorithms are in most cases concurring in their assignment of Computer Science I grades to their respective clusters.

4.2.3 Using Different Cluster Sizes

As a matter of interest, one of the experiments that were carried out was to determine the performance of different algorithms, given different numbers of clusters (K). Four different values of K (=2,3,4,5) were selected and used to compare three clustering algorithms: Simple K-means (SKM), Farthest First (FF) and Expectation-Maximization (EM). Table 4.13 shows the performance (incorrectly clustered instances) of each algorithm as the number of clusters increases.

A clear pattern is evident from these results. An increase in the number of clusters is accompanied by a corresponding increase in the number, as a percentage, of incorrectly clustered instances. This in fact amounts to a decrease in performance on the part of these algorithms. The FF algorithm has the least number of incorrectly clustered instances and EM, on the other hand, has the most incorrectly clustered number of instances. This means that FF partitions the data into clusters better than EM does. It can also be seen from Table 4.13 that on average, as the cluster size increases, so does the percentage of incorrectly clustered instances.

Table 4.11: Cluster Summary Results - Gender and Performance

Cluster	Model	A		B		C		D		F	
		Gender		Gender		Gender		Gender		Gender	
		F	**M**	**F**	**M**	**F**	**M**	**F**	**M**	**F**	**M**
0	EM	N	Y	N	Y	N	Y	N	Y	N	N
	FF	N	Y	N	Y	Y	Y	N	Y	N	Y
	MDBC	N	Y	Y	Y	N	Y	Y	Y	N	N
	SKM	N	Y	N	Y	N	Y	Y	Y	N	N
1	EM	N	N	N	N	N	N	N	N	N	Y
	FF	N	N	N	N	y	N	Y	y	y	N
	MDBC	N	Y	N	Y	N	Y	N	N	y	N
	SKM	N	Y	N	y	N	Y	N	N	y	Y
2	EM	N	N	y	N	Y	N	N	N	Y	N
	FF	N	Y	N	y	N	N	N	y	N	Y
	MDBC	N	N	N	N	Y	N	N	N	Y	N
	SKM	N	N	y	N	Y	N	N	N	Y	N
3	EM	N	N	N	N	N	N	Y	N	N	N
	FF	N	N	y	N	N	N	N	N	Y	N
	MDBC	N	N	N	N	N	N	N	N	N	Y
	SKM	N	N	N	N	N	N	N	N	N	Y
4	EM	Y	Y	N	Y	N	Y	N	N	N	N
	FF	y	N	N	N	y	N	N	N	N	N
	MDBC	y	N	N	N	y	N	N	N	N	N
	SKM	y	y	N	N	y	N	y	N	N	N

More information was obtained from the clustering experiments that were performed, particularly focusing on race and gender and their relation to academic performance in Computer Science I. Two cluster sizes were considered (K=2 and K=5) and three clustering algorithms (SKM, FF and EM) were tested to determine how they would perform, given these two cluster sizes. The results are presented in Table 4.14 and Table 4.15, respectively.

Looking at Table 4.14, it can be seen that both SKM and EM partition the given dataset into clusters of equal size which, however, do not contain the same information on race, gender and performance in Computer Science I. In the table, SKM identifies Black male students as passes (cluster 0) and Black female students as failures (cluster 1). EM also identifies Black males as passing Computer Science I and, similar to SKM, it assigns their records to cluster 0. However, EM identifies the data instances in cluster 1 with White male students who do not complete Computer Science I, as opposed to Black female students who fail Computer Science I. The FF algorithm produces different cluster sizes from SKM and EM. There are 90 data instances in cluster 0 and 10 in cluster 1. Like SKM and EM, the members in cluster 0 are identified as successful Black male students. However, cluster 1 identifies the 10 students as Coloured females who are failing Computer Science I. Even though these results may be interesting it is not clear, in the absence of other factors, that we can make much of it.

In Table 4.15 the number of clusters has been increased to 5. SKM identifies Black males and White females as students who pass Computer Science I. This result is also supported by the data in Table 4.9, which shows that in *Cluster 0*, the SKM algorithm is identifying Black students as passes by placing a y or a Y under each 'B' column for each PAS grade. Looking at *Cluster 4* a y can be seen under column 'W', indicating one or more White students who have passed Computer Science I with a distinction.

Table 4.12: EM,MDBC,SKM and FF Clustering Results - Gender and Performance
Computer Science I Grade

Cluster	A	B	C	D	F
0	**All**	(EM,FF,SKM) MDBC	*(EM,MDBC,SKM)* FF	(EM,FF) (MDBC,SKM)	*(EM,MDBC,SKM)* FF
1	(EM,FF) (MDBC,SKM)	(EM,FF) (MDBC,SKM)	(MDBC,SKM) EM FF	*(EM,MDBC,SKM)* FF	(FF,MDBC) EM SKM
2	*(EM,MDBC,SKM)* FF	(EM,SKM) FF MDBC	*(EM,MDBC,SKM)* FF	*(EM,MDBC,SKM)* FF (close)	*(EM,MDBC,SKM)* FF
3	**All**	*(EM,MDBC,SKM)* FF	**All**	(FF,MDBC,SKM) EM	(MDBC,SKM) EM FF
4	(FF,MDBC) EM SKM	(FF,MDBC,SKM) EM	(FF,MDBC,SKM) EM	(EM,FF,MDBC) SKM (close)	**All**

Table 4.13: Incorrectly Clustered Instances Using Different Cluster Sizes
Clustering Algorithms

Cluster Number (K)	SKM (%)	FF (%)	EM (%)	Average (%)
2	43.2916	37.8354	48.4794	43.2
3	56.5295	51.0733	61.4490	56.4
4	63.0590	51.0733	64.5796	59.6
5	63.0590	57.5134	64.8479	61.8
Average	56.4848	49.3739	59.8390	

Table 4.14: Using Cluster Size K=2

	SKM			**FF**			**EM**		
Cluster	Size	Race,Gender	Grade	Size	Race,Gender	Grade	Size	Race,Gender	Grade
0	74	Black, M	PAS	90	Black,M	PAS	74	Black,M	PAS
1	26	Black, F	FAL	10	Coloured, F	FAL	26	White, M	WNC

Table 4.15: Using Cluster Size K=5

	SKM			**FF**			**EM**		
Cluster	Size (%)	Race,Gender	Grade	Size (%)	Race,Gender	Grade	Size (%)	Race,Gender	Grade
0	46	Black, M	PAS	61	Black, M	PAS	22	White, M	WNC
1	20	Black, F	FAL	5	Coloured, F	WSU	44	Black, M	PAS
2	20	White, M	WNC	20	White, M	WSU	12	Indian, M	FSB
3	8	Indian, M	FAB	5	White, F	PDS	17	Black, F	FAL
4	5	White, F	PDS	8	Indian, M	FAL	5	White, F	PDS

Table 4.11 shows these to be Black male and White female students. In *Cluster 0* of this table, across the SKM row and under the 'M' column, the symbol Y is used to indicate that in this cluster the majority of passes are male. Looking at *Cluster 4*, it can be seen that in the SKM row and under the 'F' column, female students obtain distinctions. These are White students, as shown in Table 4.9. A number of Black females are failing Computer Science I. Looking at the SKM row of *Cluster 2* and *Cluster 3* of each

of the two tables (Table 4.9 and Table 4.11), under the 'F' column, it is indeed clear that some Black females are not succeeding in Computer Science I. Some White male students do not complete their studies. Looking at *Cluster 1* in Table 4.9 and Table 4.11, in the SKM row, under column 'F', it can be seen that White male students are identified as `Y` under column 'F', which indicates their failure due to not completing Computer Science I. Increasing the number of clusters has also highlighted the presence of Indian students in the Computer Science I class. The results indicate that Indian males are mostly absent from their examinations, as a result of which they fail Computer Science I. Again, this can be confimed confirmed by looking at Table 4.9 (*Cluster 1, Cluster 2* and *Cluster 4*) and Table 4.11 (Cluster 1 and *Cluster 3*). Results from the Farthest First algorithm are consistent with those obtained by SKM, which show Black males and White females as passing Computer Science I. They also show male Indian students to be failing Computer Science I (See *Cluster 0* in Table 4.9 and Table 4.11). The difference between them is the absence of a SKM cluster consisting of Coloured students.

4.3 Relationship between a module perceived to be difficult and performance in the overall Computer Science I course

The following question was posed to look into this issue in some depth:

What is the effect of the LoC module on overall performance in Computer Science I?

In most curricula there are some courses or modules that, for a number of reasons, are considered to be difficult. Some of the reasons include content difficulty, teaching level, a high failure rate or a low final mark average. The first year Computer Science course at Wits University consists of four modules: Basic Computer Organization (BCO), Fundamental Algorithmic Concepts (FAC), Data and Data Structures (DDS) and Limits of Computation (LoC). Of these, LoC is considered by a number of Computer Science I students to be the most difficult module. LoC focuses primarily on the following three areas: computability, artificial intelligence, social and ethical issues. Amongst other topics, students learn about Turing machines, the Halting Problem, models of computation, topics in artificial intelligence, ethics and so on. One of the aims of this research, therefore, was to determine whether the LoC module, perceived by some students to be difficult, influenced performance in Computer Science I.

In order to try and answer the question above, the following approach was taken. Subsection 4.3.1 sets the context for a better understanding of the LoC module, by showing its performance in relation to other Computer Science I modules, including the Computer Science I unit. Subsection 4.3.2 presents the results obtained from applying the three selected classifiers (J48, DT and Naive Bayes) on the training data set. Subsection 4.3.3 presents the results obtained when the classifier models produced from the training phase were applied on the test dataset consisting of LoC marks.

4.3.1 Performance in Computer Science I

Table 4.16 shows the performance of Computer Science I students over a 3-year period (2010-2012). The following statistics describe the results presented Table 4.16:

The last row in the last column in Table 4.16 shows clearly that compared to other modules, the average mark in LoC is the lowest. Furthermore, it can be seen that the highest number of failures occur in this module. This lends some support to the view or perception held by students and other role players within the school that LoC is the most difficult of the four Computer Science I modules. Apart from 2010, when the LoC mark was the highest compared to the other modules, the average from 2011 on-wards was the

`A` The average mark in Computer Science I (`CS-1`), including each module.

`NF` The number of failures in `CS-1`, including each module.

`MM` The maximum or highest mark obtained in `CS-1`, including each module. Bonus marks are sometimes awarded, which may result in marks that exceed 100.0%

`mM` The minimum or the lowest mark obtained in `CS-1`, including each module.

Table 4.16: Performance in Computer Science I Modules

	2010				**2011**				**2012**				**2010-2012**			
Subject	A	NF	MM	mM	A	NF	MM	mM	A	NF	MM	mM	A	NF	MM	mM
CS-1	60.8	27	98	4	57.2	31	90	0	58	33	94	4				0
BCO	**59.3**	39	97	2	62.5	25	98.2	0	58	33	94	4	58.9	97	98.2	0
DDS	60.1	52	101	10	59.4	36	96.9	0	56.7	38	89	0	58.9	126	101	0
FAC	60.8	28	97	0	61.2	14	88.0	0	56.9	25	84	0	59.7	67	97	0
LoC	63.0	26	103.1	0	**45.9**	80	79.4	0	**56.5**	34	91	9	**55.9**	**140**	103.1	0

worst. During these three years, LoC was taught by three different lecturers. The lecturer who taught it in 2011 is the one who has normally been assigned to teach it in previous years, before 2010.

4.3.2 Classifiers - Selection, Training and Comparison

The approach towards answering the question in this section was to see how data mining techniques, more specifically classification techniques, could be used to determine whether LoC could predict performance in Computer Science I. Three classifiers were selected and used: J48, Decision Table and Naïve Bayes. These and other classifiers are available in the WEKA toolkit. Each of the algorithms was applied on a training dataset consisting of 301 records. The training dataset had two variables: the LoC marks and the Computer Science I grade. The LoC mark was a numeric value while the `CS-1` grade had a nominal value. These nominal values had the classes: pas (pass), fal (fail) and pds (pass with distinction). The 10-fold cross validation testing mode was used. Table 4.17 shows how the three classifiers performed on the given training set.

Table 4.17: Classifier Performance Using Training Set (2010-2011)

	Algorithm		
	J48	DT	NB
Correct	77.41	75.75	78.07
Incorrect	22.59	24.26	21.93

Table 4.18: Difference between observed agreement and expected agreement

Kappa Value	**Strength of agreement**
<0	poor
0-0.20	slight
0.21-0.40	fair
0.41-0.60	moderate
0.61-0.80	substantial
0.81-1.00	almost perfect

Table 4.17 shows the Naïve Bayes classifier as being the best performer amongst the three. It has correctly classified most of the instances in the training set. Its kappa statistic value is the highest at 0.5093, followed by J48 and Decision Table whose kappa values are 0.4783 and 0.4593, respectively. The kappa statistic is computed as follows [Viera and Garette 2005]:

$$kappa(K) = \frac{p_o - p_e}{1 - p_e} \tag{4.1}$$

where

- p_o is the observed agreement which is 0 if there are no disagreements and 1 if there no agreements.
- p_e is the expected agreement. It is calculated by finding the probability of random agreement between the two variables.

The kappa statistic is a measure of the strength of agreement between two observations [Landis and Koch 1977]. Kappa values between 0 and 1 are an indication that the agreement between two observations of the same phenomenon are not due to chance. A kappa value of 0 indicates chance agreement, whilst a value of 1 indicates perfect agreement. The kappa values obtained in the experiments we conducted indicate clearly that the agreement between actual (observed) and expected measures is not due to chance. Using the information in Table 4.18, it can be seen that the kappa values for Naïve Bayess, J48 and DT indicate moderate agreement, with Nave Bayes performing better compared to the other two classifiers. From this result it can be concluded that, given the LoC mark of a Computer Science I student, it can be predicted with high accuracy whether the student will pass or fail Computer Science I. The results show that all three algorithms were able to correctly predict more than 75% of the instances , an indication that using the selected classification techniques, the LoC scores can be used to predict academic performance in the Computer Science I unit. For each classifier, its confusion matrix is shown in Table 4.19. All three classifiers were considered suitable for use in the next phase, the testing phase, where they were applied on the test data set.

Table 4.19: Confusion matrices for J48,NB and DT

		J48					**NB**					**DT**		
		Predicted Class					Predicted Class					Predicted Class		
		pas	**fal**	**pds**			**pas**	**fal**	**pds**			**pas**	**fal**	**pds**
	pas	187	4	5		**pas**	184	4	8		**pas**	180	4	12
Actual	**fal**	31	34	0	Actual	**fal**	31	34	0	Actual	**fal**	31	34	0
class	**pds**	28	0	12	class	**pds**	23	0	17	class	**pds**	26	0	14

4.3.3 LOC Test Dataset - Predicting Computer Science I Performance

The models produced during the training phase were applied on the LoC test dataset. The test dataset consisted of a numeric value in the LoC mark variable and a variable with an unknown value with a nominal type. This nominal type is of the same class as described in the section 4.3.2. This was to see whether the selected data classification methods, applied on the LoC dataset, could predict academic performance in the Computer Science I unit. A total of 125 LoC test scores were used. Table 4.20 shows the results obtained when the selected were applied on the test data set.

The Decision Table and J48 classifiers show similar, good performance in predicting the final, Computer Science I grades from the test dataset. Either one of these two algorithms could be used for predicting performance in Computer Science I, using LoC marks. Decision Table and J48 are more accurate compared to Naïve Bayes, whose accuracy is 86.4%. However, given the fact that its accuracy is also quite

Table 4.20: Performance of NB, J48 and DT on the Test Dataset

Classifier Performance	NB	J48	DT
Correctly Classified Instances:	86.4%	89.6%	89.6%
Incorrectly Classified Instances:	13.6%	10.4%	10.4%
Kappa statistic:	0.5193	0.6104	0.6104
Specificity:	0.55	0.59	0.59
False Positive Rate (FPR):	0.45	0.409	0.409
Mean absolute error:	0.0767	0.0812	0.0919
Root mean squared error:	0.179	0.1837	0.1864
Total Number of Instances:	125	125	125

good, Naïve Bayes can also be used for the task of predicting the Computer Science I mark from LoC scores. The fact that all three classifiers performed so well, by generating a low percentage of incorrectly classified instances, indicates that LoC may be considered as one of the factors that may be influencing performance in Computer Science I. Please note that the Error values are not important for classification tasks, they are more meaningful in the regression modeling [Bouckaert *et al.* 2013].

4.4 Performance Prediction Using the Mid-year Mark

The following question was posed in order to test the idea that the mid-year mark obtained from two modules could be used to predict performance in Computer Science I:

> *How effectively and accurately do classification methods predict overall performance in Computer Science I, when applied on the first semester performance scores of two Computer Science I modules?*

In the school, Computer Science I is taught in two semesters. At the end of the first semester students write final examinations in two of the four modules. By the time the second semester begins, every Computer Science I student will be knowing how well or how poorly they have performed in the first semester. For some of those whose performance has not been good in the first semester, there might be some fear, lack of confidence and even anxiety that they might also not do well in the second semester, leading to them failing Computer Science I. As a school, one of the most important goals is to improve the first year pass rate. Being able to predict how students are going to perform at the end of the academic year, using their first semester (mid-year mark), is of utmost importance. Timely prediction may help to decrease the failure rate, and intervention strategies may be put in place to help struggling students in the second semester, in order to help them improve their chances of success at the end of the academic year.

Focusing on the mid-year mark to predict the final year-end mark using classification techniques has not been explored to lengths, as far as the literature reveals. This section, presents the results when using the first semester/mid-year mark to predict the final year-end Computer Science I mark. The first subsection, 4.4.1, looks at comparing the three classifiers using the training dataset. This not only presents the better performing classifier but also shows the influence of the mid-year mark on the final year-end Computer Science I mark.The latter half of the section, looks at the tool created for the prediction and the results of the test phase of the modelling. This will highlight the accuracy that came from attempting to predict the final year-end mark using the mid-year mark.

4.4.1 Comparison of classifiers

Predictive models were obtained using the training dataset (data from the years 2010 to 2011). The results procured in the training phase of the modeling are presented in Table 4.21. The tree from the J48 Classifier and the rules from the Decision Table are shown in Figure 4.5 and 4.6 respectively.

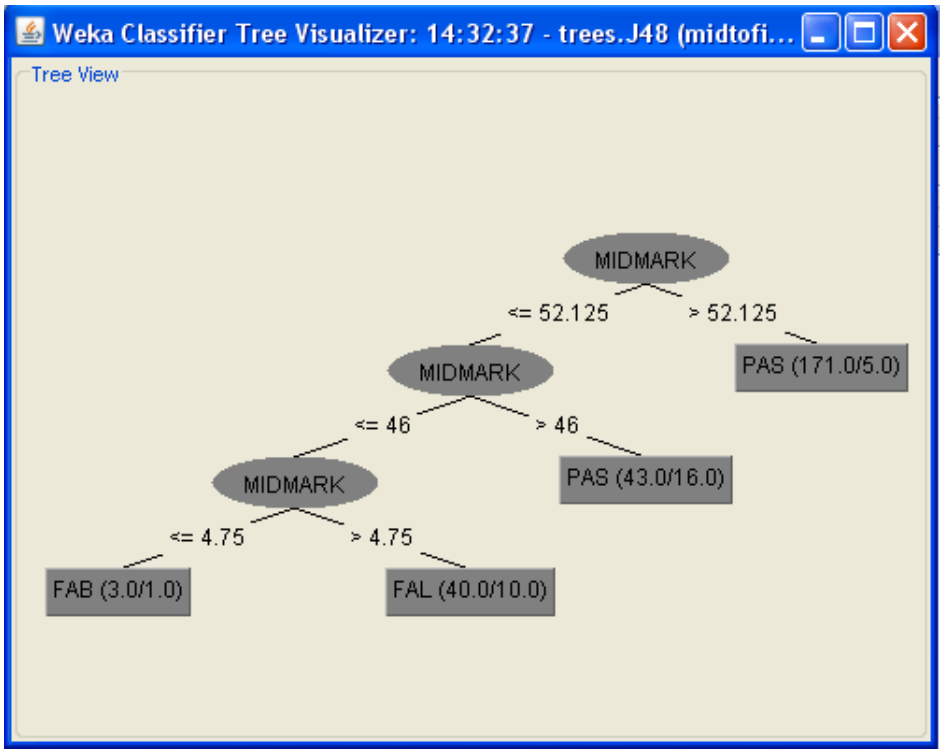


Figure 4.5: Classification Tree from the J48 model

```
Rules:
=====================================
MIDMARK          ComsGrade
=====================================
'(-inf-6.625]'   FAB
'(6.625-46.25]'  FAL
'(46.25-52.2]'   PAS
'(52.2-inf)'     PAS
=====================================
```

Figure 4.6: Rules from the Decision Table

The rules presented in Figure 4.6 on performance using the midyear mark (MIDMARK) can be interpreted as follows:

- If the mark obtained is greater than -inf and less than or equal to 6.625%, the predicted outcome is Fail, absent from examination (FAB). The student is more likely to have missed an examination in one or both of the modules that contribute to the midyear mark.
- If the mark obtained is greater than 6.625% and less than or equal to 46.25%, the predicted outcome is Fail (FAL). The student is likely to fail Computer Science I.
- If the mark obtained is greater than 46.25%, the predicted outcome is a Pass (PAS). The student is likely to pass Computer Science I.

The J48 classifier performed better with 86.38% of the instances being correctly identified in comparison to the 80.54% and 84.05% of the Decision Table and Naïve Bayes respectively (Table 4.21). There was no difference in the time each algorithm took to build the models, therefore that cannot be considered.

Table 4.21: Performance of Classifiers for 2010-2011

Algorithm	Correct (%)	Incorrect (%)
J48	86.38	13.62
Decision Table	73.61	26.39
Naïve Bayes	72.47	27.53

Furthermore, it was necessary to use performance measures to evaluate the performance of each algorithm. The Decision Table performed better than the other two algorithms in the specificity, precision and false positive rate as shown in Table 4.22 [Landis and Koch 1977]. In the educational environment, one of the key focuses is identifying students at risk of failing. This would mean that an algorithm that is better able to identify the negative instances correctly would be ideal. Since the Decision Table algorithm outperformed the J48 Classifier and Naïve Bayes with regards to higher specificity.

Table 4.22: Performance Measures

	J48	Decision Table	Naïve Bayes
Sensitivity	0.9602	0.9194	**0.9701**
Specificity	0.5714	**0.7321**	0.4643
Precision	0.8894	**0.9194**	0.8667
Accuracy	**0.8755**	0.8249	0.8599
False Positive Rate	0.4286	**0.2679**	0.5357

This meant that both the J48 classifier and the Decision Table were competitors for the better performing classification algorithm. For this reason, the run information from WEKA was investigated further. This tells us that the J48 Classifier has the highest Kappa statistic, at 0.5616, than the Decision Table and Naïve Bayes. Furthermore the J48 classifier achieved the lowest values for the mean absolute error (0.0668) and the root mean squared error (0.1919). The mean absolute error (MAE) is the average of the absolute values determined by calculating the differences between predicted and actual (observed) values. The root mean squared error (RMSE) is regarded as the standard deviation of the differences between predicted and actual values. For a particular classifier, the lower the MAE and the RMSE values, the better.

For these reasons the J48 Classifier is selected as the algorithm of choice for the prediction of the final `COMS1000` grade from the 1st semester mark and used in the SOFD tool.

4.4.2 Prediction using SOFD Tool

This investigation included, but was not limited to, using the 1st semester mark to predict the final `COMS1000` mark. The second phase of the modelling is the testing. Using the SOFD tool, the mid year mark test dataset (2012) was classified using the J48 model built from the 2010-2011 (training) dataset. The results are shown in Table 4.23.

92.54% of the `COMS1000` students' final grades were predicted correctly from the first semester mark. From a dataset of 134 students, 25 of these students failed Computer Science at the end of the year. 23 of these students were correctly predicted, this would mean not all the "students at risk" were flagged but an intervention could have occurred for 92% of these students. Bayer *et al.* [2012] suggested that a method which tries to intervene with the dropping out of students, or in this case finding the "students at risk", should have minimum false negatives. When academic performance in assessments in the first

Table 4.23: J48 Results

J48 Classifier Results	
Correctly Classified Instances:	92.5373%
Incorrectly Classified Instances:	7.4627%
Kappa statistic:	0.777
Mean absolute error:	0.0361
Root mean squared error:	0.1218
Total Number of Instances:	134

half of the first semester was compared with performance in the final examination, it was found that the correlation was poor and had no significance [O'Byrne *et al.* 2009]. The difference with the O'Byrne *et al.* [2009] study and this present study is that we look at the final first semester mark and not correlation with assessments within the semester. Additionally, the present study uses classification for prediction purposes and a scatter plot just to illustrate the significance of the correlation.

4.5 Influence of Mathematics I on Computer Science I

Given the need to try and understand the relationship between Mathematics I and Computer Science I, the following question was posed:

> *What is the nature of the relationship between the performance scores in Mathematics I and Computer Science I, and their respective modules?*

To be able to answer this question, the following targeted sub-questions were posed:

- *How strong is the relationship between student performance in the two units, Computer Science I and Mathematics I?*
- *How strong is the relationship between performance scores in Computer Science I and that in the two Mathematics I modules, Algebra I and Calculus I?*
- *How strong is the relationship between performance scores in the four Computer Science I modules and that in Mathematics I?*
- *How strong is the relationship between performance scores in the individual Computer Science I modules and that in the individual Mathematics I modules?*

There is a great deal of interest amongst academics and researchers in the relationship between Mathematics and disciplines such as Computer Science, Information Technology, the various branches of Engineering, and others. There is a strong emphasis on Mathematics in the Computer Science curriculum offered by School of Computer Science. To major in Computer Science, students must have passed their second year of Mathematics. In the School of Computer Science context a detailed investigation of the link between Mathematics and Computer Science has not been done. Furthermore, a significant number of the first year and second year students do not perform well in Mathematics which, given its status as a co-requisite and pre-requisite subject for Computer Science, results in these students not being able to complete their degrees in the stipulated period. This study focused on the relationship between Mathematics I and Computer Science I, and their respective modules. Mathematics I consists of *two*

modules, Algebra and Calculus. Computer Science comprises *four* modules: Basic Computer Organization (BCO), Data and Data Structures (DDS), Fundamentals of Algorithmic Concepts (FAC) and Limits of Computation (LoC).

To obtain answers to these sub-questions, and thereby provide the answer to the main question, a correlation analysis was performed to determine the strength of the relationship between Mathematics I and Computer Science I , including their respective modules. The correlation coefficient values listed in Table 2.4 were used to guide the interpretation of the correlation results obtained from the tests that were carried out [Taylor 1990].

4.5.1 The Relationship between Mathematics I and Computer Science I

The final marks obtained in Computer Science I and Mathematics I were used to determine the strength and significance of the relationship between these two courses. First, to determine the strength of the relationship, a correlation analysis test was performed. A moderately strong and positive correlation ratio of 0.638 between Mathematics I and Computer Science I was obtained. Thereafter a two-tailed t-test was performed to determine the significance of the relationship between these two courses. The following calculation was used to determine the test statistic [Spiegel *et al.* 2009]:

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \tag{4.2}$$

The relationship between Mathematics I and Computer Science I was found to be statistically significant (α=0.05, r=0.638, n=129, df=127, t-value=9.338). To confirm this result, a two-tailed z-test was conducted using the final marks in Computer Science I and Mathematics I. The means of the two courses were compared, the null hypothesis being that there was no difference between them (see "Hypothesized Mean" row in Table 4.24). The two-tailed z-test is normally performed when dealing with large sample sizes (n$\geq$30). Table 4.24 summarizes the results obtained from this test.

Table 4.24: z-test - Mathematics I and Computer Science I

	Mathematics I	Computer Science I
Mean	49.968	56.677
Standard Error	1.213	1.343
Standard Deviation	13.566	15.492
Standard Error of Estimate	10.492	10.492
Known Variance	184.047	239.993
Observation	125	133
Hypothesized Mean	0	
z (test statistic)	-3.706	
p(Z$\leq$z) one-tail	0.0001	
z Critical one-tail	1.645	
p(Z$\leq$z) two-tail	0.0002	
z Critical two-tail	1.960	

These results confirm the fact that the relationship between Computer Science I and Mathematics I is a significant (α=0.05, p-value=0.0002).

Mathematics I and Computer Science I - A Data Mining Perspective

Another important area of focus in the relationship between Mathematics I and Computer Science I was to look at Mathematics I as a possible predictor of performance in Computer Science I, and to determine how accurate a predictor it would be. To find the answer to this part of the question, it was decided to use the linear regression modelling technique. Put simply, its objective is to predict the value of a response variable `y`, given the value of another variable `x`. The relationship between `x` and `y` is expressed in the form of a *regression equation*.

Regression analysis experiments were performed in two phases. In the first phase (training) numeric performance scores obtained in Mathematics I and Computer Science I were used to create a regression model. In the second phase (testing), the model was used to predict performance in Computer Science I, given the test dataset of Mathematics I scores.

Descriptive Summary

A total of 609 records were used to create a regression model from the training dataset. The model was created using the final performance scores (numeric) in Mathematics I and Computer Science I, the details of which are summarized in Figure 4.7 and 4.8.

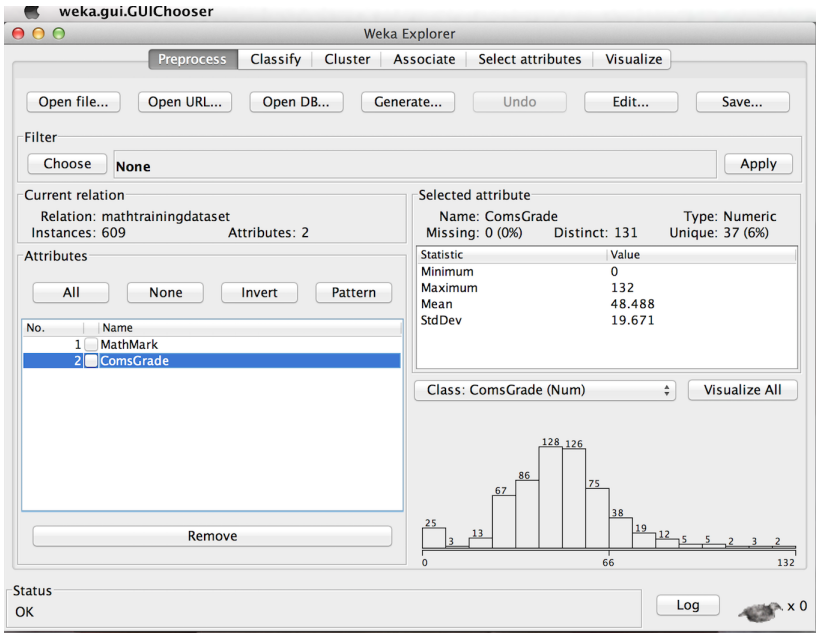


Figure 4.7: COMS1000 Visualisation

About the data:

- The 609 records were stored in a WEKA file, `mathtrainingdataset.arrf`.
- Two attributes were used: `MathMark` and `ComsMark`.

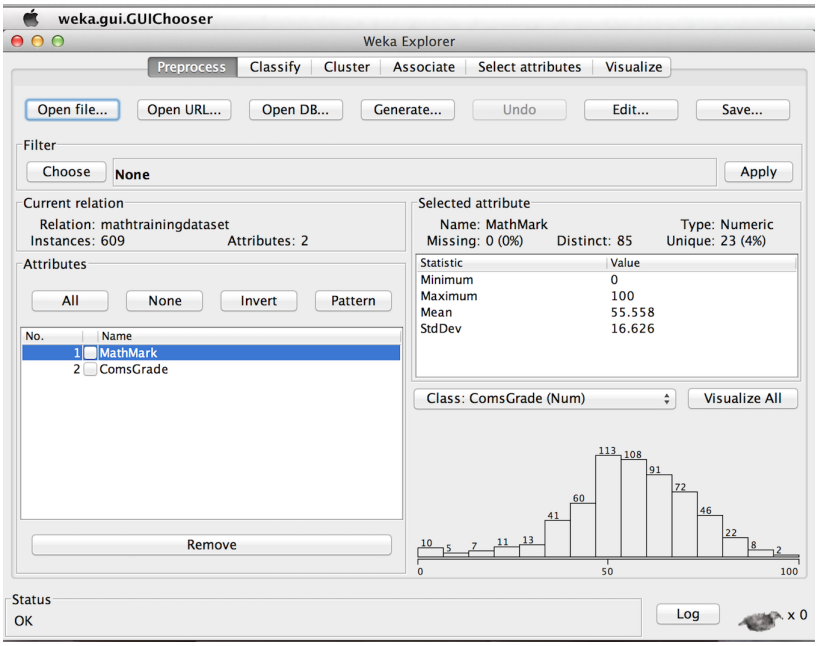


Figure 4.8: Math I Visualisation

Training Phase - Creating the Regression Model

To create the regression model, the training dataset (`mathtrainingdataset.arrf`) was loaded into WEKA, from which the Linear Regression algorithm was selected and applied on the training dataset, by choosing WEKA's *10-fold cross-validation* as the testing mode. The results obtained during this phase can be seen in Figure 4.9.

The model produced by applying the LinearRegression algorithm on the training dataset is given by Equation 4.3

$$ComsMark = 0.6046 * MathMark + 14.8988 \tag{4.3}$$

where

- the y-intercept is b_0=14.8988
- the slope is b_1=0.6046
- `MathMark` is the predictor variable (`x`) and `ComsMark` is response variable (`y`)

The value of the slope is interpreted as follows: an increase of one unit (point) in the Mathematics I mark is accompanied by a corresponding increase of 0.6046 in the Computer Science I mark. Since it is possible to obtain a zero mark in Mathematics, the y-intercept value of 14.8988 would be the estimated `ComsMark` a Computer Science I student would be expected to get if she or he obtained a score of zero in Mathematics I. That is, such a student would also be expected to fail Computer Science I.

The value of the *correlation coefficient* is r=0.5037. It indicates the strength and the direction of the relationship between the `ComsMark` and `MathMark` variables. The relationship between these variables is moderately strong [Taylor 1990]. A positive relationship can be taken to mean that an increase in a Mathematics I score is accompanied by a corresponding increase in a Computer Science I score. In other words, there is a *direct relationship* between the two subjects. Another way of looking at the strength of the relationship between two variables is to use the *coefficient of determination* (r^2), a measure that indicates the amount of variability in the response variable, which is accounted for by its relationship

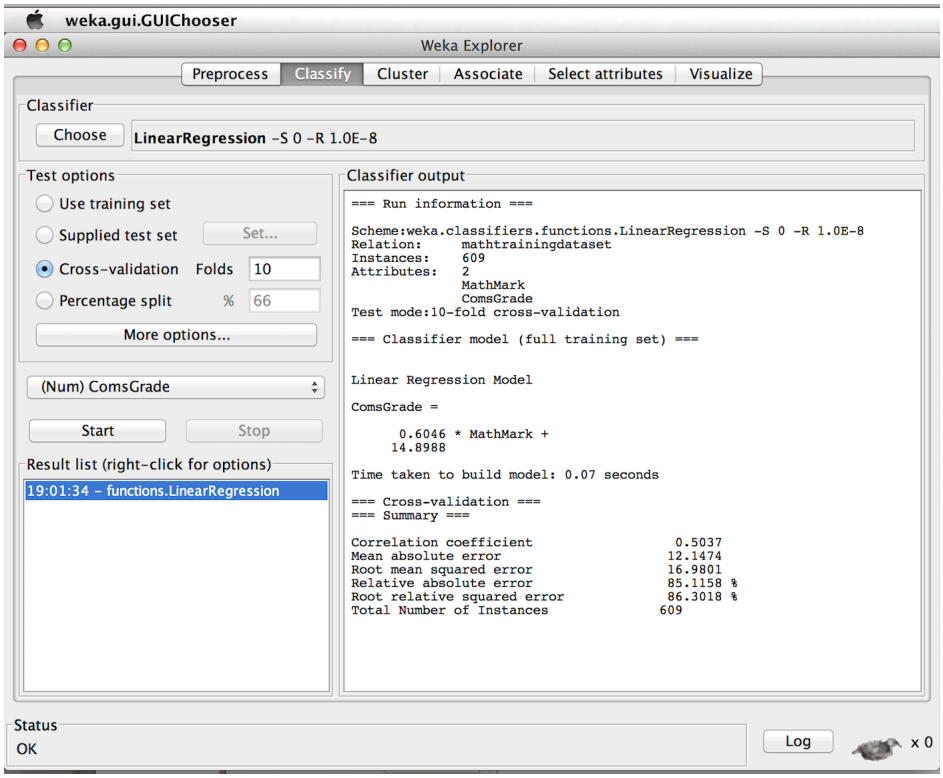


Figure 4.9: Linear Regression Results (Training dataset)

to another variable. Using the value of the correlation coefficient, the coefficient of determination was determined (r^2=0.2537, $\approx$ 25%). What this value means is that 25% of the total variation in Computer Science I performance is influenced by Mathematics I. The remaining 75% is accounted for by other, unknown factors, such as gender, race, high school attended, prior programming experience, high school mathematics mark, etc.

Testing Phase - Predicting Computer Science I Scores

The regression model produced during the training phase was tested by applying on the 2012 test dataset of Mathematics I scores. The actual scores in Computer Science I were already known. The aim during this phase was to predict Computer Science I scores using the model generated during the training phase, and to compare the predicted scores to the actual scores. The 113 records in the test dataset were stored in a WEKA file, `mathdataset.arff`. Figure 4.10 shows the results when the model was applied on the test dataset.

The correlation coefficient between Computer Science I and Mathematics I is r=0.7899. This result means that

- the relationship between the performance scores in Computer Science I and those in Mathematics I is a strong one,
- students who obtain high marks in Mathematics I can be expected to obtain high marks in Computer Science I. This is not being said to imply that there is a *causal* link between Mathematics I and Computer Science I

```
=== Re-evaluation on test set ===

User supplied test set
Relation:     mathdataset
Instances:     unknown (yet). Reading incrementally
Attributes:   2

=== Summary ===

Correlation coefficient                  0.7899
Mean absolute error                      8.3069
Root mean squared error                 10.6376
Total Number of Instances              113
```

Figure 4.10: Linear Regression Results (Test dataset)

- if we know the value of a score in Mathematics I, the corresponding Computer Science I mark can be predicted with greater accuracy (since we have a strong r).

The coefficient of determination (r^2) was obtained by calculating the square of the correlation coefficient (r). The coefficient of determination was calculated to be r^2=0.6239, $\approx$ 62%. This result means that 62% of the total variability in Computer Science I can be accounted for by the variability in Mathematics I. This indicates a strong association or relationship between the two subjects. It is important to note the decrease in errors in the test data modelling which plays a role in the increase in correlation coefficient. Other factors such as having less instances in the test data set could also potentially influence this. Further investigation, with regards to this, is required but is out of the scope of the current research.

The results obtained and presented in this section provide a clear indication that, from a data mining perspective, linear regression modeling can be used for the task of predicting Computer Science I scores from Mathematics I scores.

4.5.2 Relationship Between Mathematics I Modules and Computer Science I

Mathematics I comprises two modules, Algebra and Calculus, which are taught and assessed separately by the School of Mathematics. Computer Science I students are required to take both modules together, as co-requisites of the Computer Science I course. Having looked at the question of Mathematics as a whole and its relationship with Computer Science I, it was considered equally important to determine how strongly each Mathematics I module related to Computer Science I. Using the final marks obtained in each module, and the final marks in Computer Science I, a correlation test was performed between each of the modules and Computer Science I. A summary of the results obtained is shown in Table 4.25.

There is a positive correlation between Algebra and Calculus, on the one hand, and Computer Science I, on the other. In both cases, the nature of the correlation can be categorized as being moderately strong. For Algebra the p-value (0.173) for the two-tailed test is much greater than the specified level of significance (α=0.05). Furthermore, the value of the test statistic (z-value) is -1.362, which (even as an absolute value) is less than the critical value of 1,96. This further confirms the fact that Algebra and

Table 4.25: Correlation of Mathematics I Modules with Computer Science I

	Algebra I	Calculus I
Observations (n)	125	125
Correlation (r)	0.624	0.584*
t-value	9.004	8.105
z-value (test statistic)	-1.362	-5.085
p(Z≤z) two-tail	0.173	0.000
z Critical two-tail	1.960	1.960

Computer Science I, even though moderately strongly correlated with each other, the correlation between them is not significant. On the other hand, for Calculus the p-value is smaller compared to the level of significance ($\alpha = 0.05$), and also the z-value (test statistic) is greater than the critical value of 1.960, both of which facts indicate a significant relationship between Calculus and Computer Science I.

4.5.3 Relationship Between Mathematics I and Computer Science I Modules

Another question this research sought to answer was about the strength of the relationship between Mathematics I, on the one hand, and each of the four Computer Science I modules, on the other. A correlation analysis test was performed, to determine the relationship between the final scores obtained by Computer Science I students in Mathematics I and the final performance scores in each of the modules of Computer Science I. The results are summarized in Table 4.26.

Table 4.26: Correlation of Mathematics I with Individual Computer Science I Modules

	BCO	DDS	FAC	LOC
Size (n)	132	133	129	126
Corr. (r)	0.54*	0.42*	0.73*	0.58*
z-value	-4.27	-3.97	-5.05	-3.56
p(Z≤z)	0.00	0.00	0.00	0.00
z Crit.	1.96	1.96	1.96	1.96

There is a positive and significant correlation between Mathematics I and each of the Computer Science I modules. There is a fair correlation between Mathematics I, on the one hand, and BCO and DDS, on the other. There is a moderately strong correlation between Mathematics I and the other two Computer Science I modules, FAC and LOC. Mathematics I correlates the least with DDS (0.42) and the most with FAC (0.73). The stronger correlation of Mathematics I with the FAC and LOC modules is most probably attributable to theoretical/mathematical content that forms part of the material being taught in both modules. Looking at each z-value or p-value in each module, and using a similar interpretation of these results as was done in Subsection 4.5.2, it can be seen that the individual relationships between Mathematics I and Computer Science I modules is indeed statistically significant.

4.5.4 Relationship Between Mathematics I Modules and Computer Science I Modules

It was also necessary in this research to determine the module-to-module relationships between Computer Science I and Mathematics I. The final marks in all the modules were used to perform a correlation analysis test, to determine the relationship between the two Mathematics I modules (Algebra I and Calculus I) and the four Computer Science I modules (BCO, DDS, FAC and LOC). The correlation matrix, Table 4.27, shows the results that were obtained (results of interest appear in bold). An asterisk indicates

a significant relationship between the two modules involved.

Table 4.27: Correlation between Mathematics I and Computer Science I module scores

	BCO	DDS	FAC	LOC	ALG	CAL
BCO	1.00					
DDS	0.79	1.00				
FAC	0.64	0.67	1.00			
LOC	0.71	0.68	0.74	1.00		
ALG	**0.54***	**0.45**	**0.67***	**0.55**	1.00	
CAL	**0.50***	**0.37***	**0.68***	**0.54***	0.75*	1.00

The following observations were made:

- Algebra I and Calculus I correlate positively and significantly with BCO. In each case the correlation is fair.

- Algebra I and Calculus I correlate fairly and positively with DDS. For Calculus I the correlation is significant. Even though Algebra I has a higher correlation with DDS, compared to Calculus I, its correlation with DDS is nonetheless not significant.

- Algebra I and Calculus I show a moderately strong, positive and significant correlation with FAC, which is the only Computer Science I module that correlates so strongly with Mathematics I modules.

- Algebra I and Calculus I show a fair and positive correlation with LoC. However, unlike Algebra I, Calculus I correlates significantly with LoC, despite the fact that Algebra I's correlation with LoC is slightly stronger compared to that of Calculus I.

There are a few reasons why FAC, compared to the other modules of Computer Science I, correlates so strongly with Algebra I and Calculus I. A significant content in the FAC curriculum is of a theoretical and mathematical nature. Some of the topics covered in this module include algorithm analysis, mathematical proof techniques, algorithm design, the study of well-known algorithms, graph theory, recursion and so on. Given this focus in FAC content, it is therefore not surprising to obtain such high correlation results between FAC and the two Mathematics I modules.

4.6 Relationship between Course Selection and Academic Performance

In order to find some answers to this interesting issue, the following question was posed:

> *How strong is the association between the courses selected during registration by Computer Science I students and their overall academic performance?*

The courses selected by a student during enrolment at the beginning of the academic year may or may not affect that student's performance. Depending on how courses are selected, certain course combinations could have a better influence on academic performance than others. Although course selection during registration and enrolment is left up to individual students to decide upon, the students are often assisted by academic administrators and academics, who guide them and also recommend to them which particular courses and course combinations they could consider for their study. The focus in this part of the research was to look at the different courses selected by individual Computer Science I students

during registration. There were two reasons for this. First, it was to determine and study the relationship patterns among the courses selected by these students. Second, it was to see how overall performance by Computer Science I in the first year could be linked or related to their selection of courses. The results obtained from looking into this aspect may be used as a foundation for finding ways of helping students make well-informed decisions in their choice of courses, thereby improving their chances of success in first year. In essence, this part of the research looked at which course combinations would be most useful to select in order to enable a Computer Science I student to stand a better chance of achieving overall success in the first year of study. Using the Association Rule Mining technique, the different combinations of courses selected by Computer Science I students were analysed to determine useful patterns in the relationships between them, and association rules were generated which indicated the strength or weakness of the association between different courses.

For this part of the research the following approach was taken. The datasets were processed in two phases, which are discussed in Subsection 4.6.1 and Subsection 4.6.2, respectively. In the first phase (Phase 1) the predictive Association Rule Mining (ARM) algorithm, *Predictive Apriori*, was applied on a complete dataset of the different combination of courses selected by Computer Science I students. In the second phase (Phase 2), a reduced dataset was used, after which it was subjected to analysis by the *Predictive Apriori* algorithm.

4.6.1 Phase 1: Applying the Predictive Apriori Algorithm on the Complete Dataset

Descriptive Summary

This phase began by taking a closer look at the data that was to be used for analysis. The first thing was to look for the most common combinations of first year courses preferred by Computer Science I students. During registration students who opt for Computer Science I are also advised to consider taking the first year Computational and Applied Mathematics (CAM) course. Most of the students will take this course as a recommended co-requisite of Computer Science I. Figure 4.11 shows the next batch of courses most preferred by Computer Science I students, after they have taken Mathematics I (compulsory co-requisite) and selected CAM as the first non-compulsory co-requisite course.

Looking at Figure 4.11, it can be seen that the second most preferred course is Physics, closely followed by Economics. In order of preference, the majority of Computer Science I students select the following combinations of courses when they register:

1. {Computer Science I, Mathematics I, Computational & Applied Mathematics I, Physics I}

2. {Computer Science I, Mathematics I, Computational & Applied Mathematics I, Economics I}

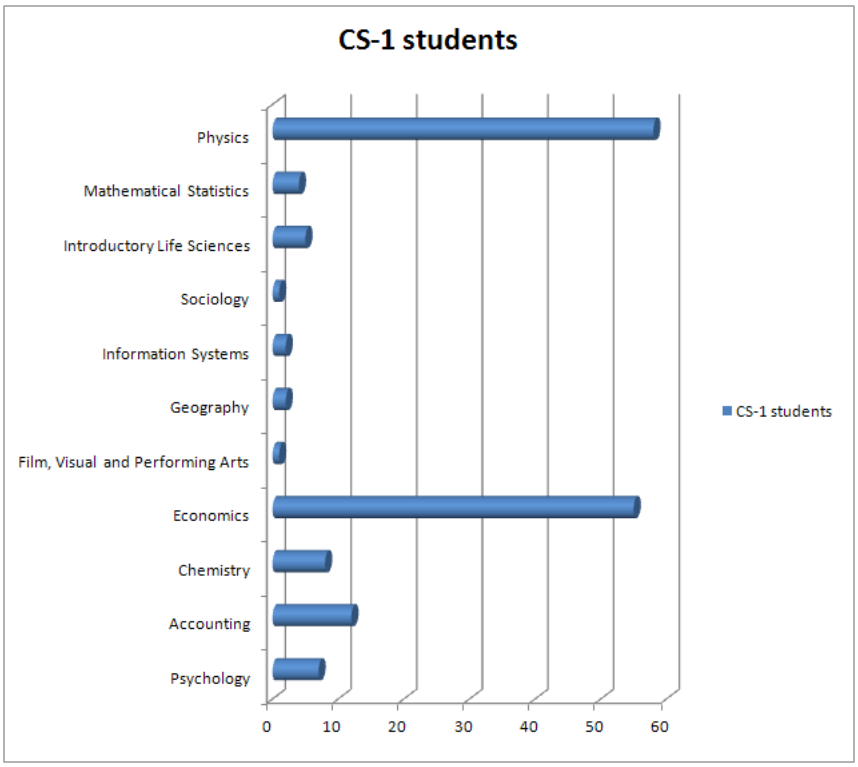


Figure 4.11: Number of Computer Science I students in different courses

Discovering Association Rules

The Predictive Apriori algorithm was applied on the full dataset and association rules were generated that depicted relationships amongst the different courses selected by Computer Science I students. The dataset contained the following attributes (Table 4.28):

Table 4.28: Selected Attributes (inputs to Predictive Apriori Algorithm)

Attribute	Code	Points
Applied Mathematics for Applied Computing I	APPM1021	18
Basic Principles of Individual & Group Psychology I	PSYC1002	18
Chemistry IA	CHEM1013	18
Computational and Applied Mathematics I	APPM1006	36
Concepts of Design I	ELEN1002	18
Economics I	ECON1000	36
Economics IIB	ECON2001	24
Information Systems IB	INFO1003	18
Introductory Life Sciences I	BIOL1000	36
Mathematical Statistics I	STAT1003	18
Physics for Applied Computing I	PHYS1023	18

The following rules, shown in Table 4.29, were generated as the best rules, by running the Predictive Apriori algorithm on the full dataset.

Each association rule in Table 4.29 has two components: an *antecedent* and a *consequent*. For example, given Rule 1,

Table 4.29: Predictive Apriori Phase 1 Results

Attribute	Best Rules Generated	Accuracy
APPM1021	1. {APPM1021=yes, INFO1003=yes} $\Longrightarrow$ RET	0.83330
PSYC1002	2. {PHYS1023=yes, INFO1003=yes} $\Longrightarrow$ RET	0.83330
CHEM1013	3. {INFO1003=yes, ELEN1002=yes} $\Longrightarrow$ RET	0.83330
APPM1006	4. {BIOL1000=yes, CHEM1013=yes} $\Longrightarrow$ PCD	0.83330
ELEN1002	5. {BIOL1000=yes} $\Longrightarrow$ PCD	0.82353
ECON1000	6. {ECON100=yes, PSYC1002=yes} $\Longrightarrow$ PCD	0.79997
ECON2001	7. {ELEN1002=yes} $\Longrightarrow$ RET	0.75004
INFO1003	8. {APPM1006=yes, STAT1003=yes} $\Longrightarrow$ RET	0.75004
BIOL1000	9. {APPM1006=yes, ECON2001=yes} $\Longrightarrow$ PCD	0.74998
STAT1003	10. {APPM1006=yes, INFO1003=yes} $\Longrightarrow$ PCD	0.74998
PHYS1023		

$$\{\text{APPM1021=yes, INFO1003=yes}\} \Longrightarrow \text{RET}$$

its antecedent is {APPM1021=yes, INFO1003=yes}, and its consequent is RET. Two metrics, *support* and *confidence*, are used to determine the strength of a given rule. Consider an association rule such as X $\Longrightarrow$ Y. Its support represents the percentage of transactions in which both X and Y are present together, and its confidence represents the percentage of transactions containing X, and which also contain Y, among all the transactions in X is found. These two measures are used in the Apriori algorithm. However, the Predictive Apriori algorithm chosen for this research uses a single, *predictive accuracy* metric, in which the rules are ranked by combining the support and confidence metrics. The Apriori algorithm, on the other hand, only uses the confidence metric to rank the rules it produces.

Analyzing the Discovered Rules

Table 4.29 contains rules that, for convenience, can be described as "positive" and "negative". Rules with ``PCD" as their consequent are regarded as positive rules. For these rules, a combination of the courses selected by Computer Science I students results in a successful *overall result decision*, ``PCD", which means that a Computer Science I student would stand a better chance of passing the first year of study and proceed to the next academic year, if she or he were to select these courses. Where the consequent of a rule is a ``RET", that rule is regarded as negative. For a combination of courses that produce a ``RET", a Computer Science I student who selects a set of courses that produce such an outcome is likely to be unsuccessful in her or his first year of study, and will be required to come back to repeat her or his studies. Let us consider a student registration or enrolment session. In such an event, course administrators, coordinators and academics could use the information provided by rules such as the ones shown in Table 4.29. For example, using information obtained from the positive rules, course administrators, coordinators and academics could use this information to encourage students to select a set of courses that would most likely result in them passing their first year of study. By focusing on the negative rules, students could be advised to avoid course combinations that would most likely result in them failing their first year of study. In this research, positive and negative rules were both taken into account when trying to answer the question regarding the relationship between course selection and academic performance in first year.

Looking at Table 4.29 it is clear that, based on the first three rules, Computer Science I students who select the courses specified in these rules can be expected to repeat their first year of study. With an accuracy as high as 0.83 for each rule, this can be expected to happen. Another fact worth noticing is the common occurrence of the INFO1003 course amongst these rules, which makes it an important contributor towards the failure rate of Computer Science I students who take this course as part of their combination of first year courses.

Another interesting fact was that INFO1003 appeared in a positive rule (Rule 10), in conjunction with APPM1006. What this tells us is that there are cases where INFO1003, when combined with "carefully" selected courses, can be part of a successful course combination. In other words, a course advisor could recommend to a Computer Science I student that, in addition to Computer Science I and Mathematics I, the student could also consider taking APPM1006 and INFO1003. With the exception of Rule 9, the accuracy of Rule 10 is somewhat lower compared that of other positive rules. Rule 4 and Rule 5 show that a Computer Science I student taking either one of the following course combinations would stand a good chance of passing first year:

- Combination using Rule 4: {Computer Science I, Mathematics I, BIOL1000, CHEM1013}
- Combination using Rule 5: {Computer Science I, Mathematics I, BIOL1000}

Both rules have relatively high accuracy rates. This is in line with research in the Computational Biology and Bioinformatics field. Bioinformatics, for example, is a new discipline that brings together concepts from Computer Science and Biology.

In Rule 8 the course APPM1006 is combined with another course, STAT1003, resulting in a student who selects this combination having to return to repeat the first year. The accuracy of this rule is high (0.75), which makes it highly probable that students who select the combination

{Computer Science I, Mathematics I, APPM1006, STAT1003}

are most likely going to repeat their first year of study. This is interesting because as a recommended additional course, it would have been expected that APPM1006 would only be found in course combinations that return positive results, as is the case with Rule 9 and Rule 10. The reason could be the inclusion of STAT1003in the course combination that results in Rule 8. However, without an in-depth look at this matter, a firm conclusion cannot be drawn.

The first three rules in Table 4.29, including Rule 7, all indicate that first year students who take the courses specified in these rules are most likely to fail their first year of study. All these courses are taught as part of a *structured curriculum*, which was recently introduced by the School of Computer Science, in conjunction with other schools at Wits University. A new degree, the *BSc in the field of Applied Computing*, was created to enable students to register for courses that would be offered in this curriculum. The first group of first year students was accepted in 2011. The following first year courses are offered as a compulsory bundle to students who are accepted to do this degree (see Table 4.30).

Table 4.30: BSc Applied Computing Courses

Course Code	Course Description	Points
COMS1000	Computer Science I	36
ELEN1002	Concepts of Design I	18
INFO1003	Information Systems I	18
MATH1034	Algebra I	15
MATH1036	Calculus I	21
APPM1021	Applied Mathematics for Applied Computing I	18
PHYS1023	Physics for Applied Computing I	18

Given the high accuracy of the rules that indicate failure, in which four of the courses in Table 4.30 are included, the schools where these courses are taught might need to work together to determine where the actual areas of difficulty are, and how the situation could be remedied to enable the first year students

enrolled in this curriculum to stand a better chance of succeeding in the first year of this challenging curriculum.

4.6.2 Phase 2: Applying Predictive Apriori on a Reduced Dataset

Descriptive Summary

A reduced dataset was also used, which consisted only of those courses in which more than 10 Computer Science I students were enrolled. All the second year courses were removed, leaving only first-year courses in the dataset, which now consisted of 528 student records and 9 attributes. Figure 4.12 shows a summary of the courses that were used in this dataset.

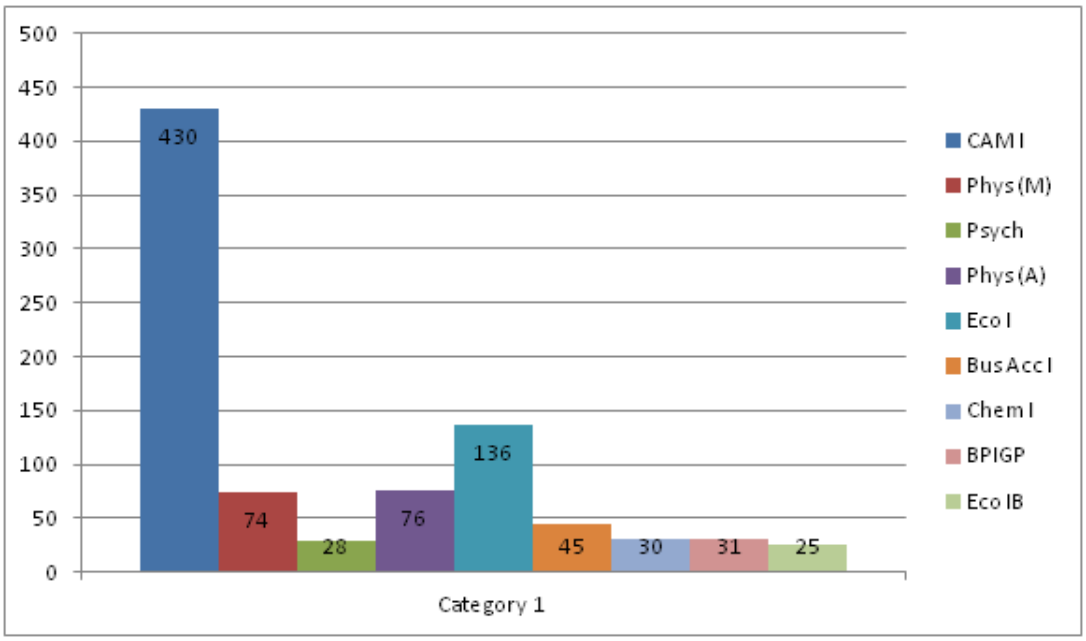


Figure 4.12: Number of Computer Science students in different courses - Phase 2 dataset

Clearly, looking at Figure 4.12, the most popular additional course among Computer Science I students is CAM I (`APPM1006`). During registration, Computer Science I students are generally advised to choose `APPM1006` as one of the co-requisite subjects of Computer Science I. This could be one of the reasons why a comparatively large number of students prefer to do this subject.

Discovering Association Rules

In this phase the Predictive Apriori algorithm was again used, and this time it was applied on the reduced dataset in order to discover more interesting rules. The results for this part of the experiment are shown in Table 4.31.

Looking at Table 4.31, the positive effect of the `BIOL1000` course is apparent. In Rule 1 and Rule 2, both of which contain `BIOL1000`, the accuracy is quite high and the rules indicate clearly that Computer Science I students who include `BIOL1000` in their course selection will most likely pass their first year of study. The combination of `ECON1000` and `PSYC1002` had a positive outcome and high accuracy

Table 4.31: Predictive Apriori Phase 2 Results

Attribute	Best Rules Generated	Accuracy
APPM1021	1. {BIOL1000=yes, CHEM1013=yes} $\Longrightarrow$ PCD	0.83333
PSYC1002	2. {BIOL1000=yes} $\Longrightarrow$ PCD	0.82353
CHEM1013	3. {ECON1000=yes, PSYC1002=yes} $\Longrightarrow$ PCD	0.80000
APPM1006	4. {ELEN1002=yes} $\Longrightarrow$ RET	0.76471
ELEN1002	5. {APPM1006=yes, STAT1003=yes} $\Longrightarrow$ PCD	0.75005
ECON1000	6. {APPM1021=yes} $\Longrightarrow$ PCD	0.73684
ECON2001	7. {PHYS1023=yes} $\Longrightarrow$ PCD	0.73684
INFO1003	8. {CHEM1013=yes} $\Longrightarrow$ PCD	0.69231
BIOL1000	9. {STAT1003=yes} $\Longrightarrow$ PCD	0.69231
STAT1003	10. {INFO1003=yes, APPM1021=yes, PHYS1023=yes, ELEN1002=yes} $\Longrightarrow$ PCD	0.25000
PHYS1023		

value, similar to the result that was obtained in Table 4.29. Generally, the majority of Computer Science I students do not select the following combination of courses when they register:

{Computer Science I, Mathematics I, BIOL1000, PSYC1002}

Given the fact that students who select this combination of courses actually pass their first year, it is an option worth looking at by course administrators and academics when they advise students about the set of courses they should select at the beginning of their first year. The study by Anderson and Moore [2009] looks at the relationship between Economics and Computer Science. The authors mention how these two subjects are closely connected in such real-life applications as information security.

From the results obtained in Phase 1 and Phase 2, it is clear that course selection does have an effect on the overall performance of a Computer Science student in the first year of study. The rules generated by the Predictive Apriori algorithm have shown clearly that there is a relationship between course selection and the academic performance of Computer Science I students in the first year of study.

4.7 Using Computer Science I to predict overall First Year Performance

In this section, the following research question is being answered:

> *Can performance in Computer Science I be used to predict overall performance in the first year of the BSc degree?*

Computer Science I is one of several courses a first-year Bachelor of Science student can register for. The selection of courses is not only limited to courses offered by the Faculty of Science. Students can also obtain permission to register for courses offered by other faculties. Within the Faculty of Science first year students normally register for a total of four courses. A student registering for Computer Science I is also required to register for Mathematics I, and then to select any other two courses of their choice. The first year of tertiary-level study, both locally and internationally, is generally characterized by a high failure rate. It is therefore important to try and identify the factor or factors that contribute to this result. From the school's perspective there was an interest to know whether performance in Computer Science I could be used to predict performance in the first year of study.

4.7.1 Prediction - Using the Training Dataset

In order to determine whether Computer Science I could be used to predict performance in the first year of the BSc degree, different classification algorithms were applied on a training dataset consisting of a total of 701 Computer Science I student records. Figure 4.13 shows a sample of the records contained in the training dataset.

```
@RELATION comstofinaltrainingdataset

@ATTRIBUTE COMSMARK NUMERIC
@ATTRIBUTE ComsGrade {PAS, FAL,PDS, FSB, WDF, WDA, FDF, FNR, FAB}
@ATTRIBUTE PROGRESSIONOUTCOME {PCD, RET, MFL, NCD, MRNM, MFC, MFE, MCE, MBZ, Q, INP, MFO, MBP, MBR

@DATA
52,PAS,RET
22,FAL,RET
54,PAS,RET
50,PAS,RET
48,FAL,RET
71,PAS,RET
42,FAL,RET
59,PAS,RET
54,PAS,RET
53,PAS,RET
57,PAS,RET
69,PAS,RET
75,PDS,RET
48,FAL,RET
67,PAS,RET
60,PAS,RET
62,PAS,RET
```

Figure 4.13: .arff File of training data set

There are three fields in each record. These are:

- `COMSMARK`, a numeric variable that stores the final Computer Science I mark.
- `ComsGrade`, a variable that categorizes a numeric mark as a *pass distinction* (`PAS`), *pass* `PAS` or *fail* (`FAL`).
- `PROGRESSIONOUTCOME`, a variable that represents the overall result decision at the end of the academic year (first year). In Figure 4.13 the code `'RET'` is used to indicate the status of a student who has not passed the first year of the BSc. degree, and who must therefore return the next academic year to repeat first year.

The training data set was classified using the J48, Decision Table and Naive Bayes algorithms. The data collected in each year, from 2006 to 2011, was classified separately. Table 4.32 summarizes the results that were obtained using the separate datasets.

The accuracy of the three classifiers is poor. Generally, across all the dataset sizes (`Sz.`), the percentage of correct instances (`Cor.`) is unacceptably low and the percentage of incorrect instances (`Inc.`) is unacceptably high. Also worth noting is the fact that the DT classifier, on smaller dataset sizes (77-117), performed better than J48 and NB. However, with the increase in dataset size (136-143), J48 outperforms DT and NB, with the latter performing the worst. None of these classifiers is at most 60% accurate, which in itself would not be considered to be a high level of accuracy.

In view of the poor classifier performance, another approach was considered, which was to combine the separate training datasets into one large dataset. The thinking behind this decision was that, given a large

Table 4.32: Classifier Performance - Computer Science I Final Outcome)

	Classifiers											
	J48				**DT**				**NB**			
Year	Sz.	Time (s)	Cor. (%)	Inc. (%)	Sz.	Time	Cor. (%)	Inc. (%)	Sz.	Time	Cor. (%)	Inc. (%)
2006	77	0.00	38.96	61.04	77	0.00	45.45	54.55	77	0.00	41.56	58.44
2007	112	0.00	54.46	45.54	112	0.00	57.14	42.86	112	0.00	55.36	44.64
2008	117	0.00	54.70	45.30	117	0.00	55.56	44.44	117	0.00	49.57	50.43
2009	136	0.00	58.53	51.47	136	0.02	53.68	46.32	136	0.00	53.68	46.32
2010	143	0.02	50.35	49.65	143	0.00	48.25	51.75	143	0.00	43.36	56.64
2011	116	0.05	45.13	54.87	116	0.02	36.28	63.72	116	0.00	43.36	56.64

dataset, the accuracy of the classifiers was most likely going to improve. Table 4.33 shows the results obtained using the combined dataset.

Table 4.33: Classifier Performance using Combined Dataset (2006-2011)

Algorithm	Time (s)	Correct (%)	Incorrect (%)
J48	0.00	45.85	54.15
Decision Table	0.02	46.56	53.44
Naïve Bayes	0.00	46.70	53.30

Looking at the results in Table 4.33, it is can be seen that overall, the performance of the classifiers is again poor, despite the fact that a large dataset was used, in the hope that a classifier such as J48 would demonstrate better performance. However, none of the classifiers are even 50% accurate. Table 4.34 shows the detailed performance of each classifier in terms of sensitivity, specificity, precision and false positive rate.

Table 4.34: Performance Measures

	J48	Decision Table	Naïve Bayes
Sensitivity	0.8837	0.9169	0.9136
Specificity	0.4635	0.4005	0.4484
Precision	0.5553	0.5370	0.5567
False Positive Rate	0.5365	0.5995	0.5516

The sensitivity values are quite high due to the small number of false negatives generated. In each case, a high fraction of relevant data records has been retrieved. The low specificity values are accounted for by the high number of false positives generated by the classifiers. The precision and false positive rate are similarly affected by the high number of false positive values. Given the generally poor performance of these classifiers, it was decided to use all of them during the testing phase.

4.7.2 Prediction - Using the Test Dataset

To predict the overall outcome in first year, the test set consisting of the final marks in Computer Science I was used. These were the marks obtained by the Computer Science I class of 2012. Each of the tree classifiers was applied on this dataset. The results are shown in Table 4.35.

Table 4.35: Performance of Classifiers for test dataset

Algorithm	Time (s)	Correct (%)	Incorrect (%)
J48	0.00	45.52	54.48
Decision Table	0.02	45.52	53.48
Naïve Bayes	0.00	46.27	53.73

There is no improvement in these results. The accuracy is still very poor. None of these classifiers is even 50% accurate. This is not to suggest that 50% would have been deemed acceptable. Given these poor results, it is clear that the final marks in Computer Science I cannot be used to predict overall performance in the first year of the BSc. degree. Since the classifiers, using the training dataset, did not produce good models, it could be expected that using these models on the test dataset was likewise not going to yield acceptable or good results. Given the poor accuracy of all the classifiers, it was not possible to convincingly answer the question that was posed; namely, that performance in Computer Science I could be used to predict overall performance in the first year of the Bachelor of Science degree.

4.8 Conclusion

The main purpose of this chapter is to present all the results obtained as an attempt to answer the six research sub-questions and therefore the main research question. Section 4.2 investigates the use of race and gender as factors to cluster CS−1 students into academic performance groups. The results show a relationship between race and gender and the academic performance in the COSM1000 unit. The second research sub-question explored a factor that has not been looked at in previous work: prediction of the COMS1000 academic performance using the module perceived to be the most difficult. Results from this section were also positive with a high level of accuracy. Similar to this, is using the mid year academic performance of the CS−1 students to predict the final Computer Science I results. And similar to the previous investigation, this factor proved to have a good correlation and therefore to be a good factor for prediction. Moving away from the COMS1000-specific curriculum related factors the next research sub-question showed results of a strong relationship between Mathematics I and Computer Science I. This investigation also showed a relationship between Mathematics and the Fundamental Algorithmic Concepts module in Computer Science I. Additionally, there was a better correlation between Algebra and Computer Science I than with Calculus. The final two research sub-questions are related to overall first-year academic performance. The first is the association of the courses that the CS−1 students select and their final progression outcome. Results show a link between the courses that the CS−1 students select and their final first-year academic performance. The last research sub-question looks at predicting the final first-year academic performance using the COMS1000 results. Results in this part of the research do not show a relationship between performance in COMS1000 and overall performance in first-year. All these results also answer the main research question because they show that the data collected and stored on the CS−1 students can give information about these students and their academic performance.

An interesting but expected result from this chapter is the good relationship between academic performance in the COMS1000 unit and three of the factors: the result in the Mathematics mark , the module perceived to be the most difficult and the COMS1000 first semester result. The last two factors are unique contributions of this research, with no found literature related to using these factors and classification for predicting academic performance. Furthermore, results presented with regards to the influence of course selections with overall academic performance showed great strides towards early interventions and the assisting of students in making the best decisions.

The following chapter, the Discussion chapter, will discuss the results presented in this chapter. The discussion chapter will compare the results found with related literature for the different secondary questions. This chapter will also look at how these secondary questions come together to answer the main research question.

Chapter 5

Discussion

Higher education institutions collect and store large volumes of data about the students they enrol. This data can be used in an attempt to produce information and patterns about these students and their academic performance. Wits University is not immune to this as it also collects data about all their students during the application and registration processes. Additionally, this data is enhanced, in the duration of the students academic career at Wits University, with data on the students' academic performance. This data, at Wits University, is occasionally used for research but could have the potential to work for the students and university. The purpose of this research is to explore this, specifically in the School of Computer Science. The main question which this research sought to answer is:

> *Can the historical data collected at Wits University about first year Computer Science (CS-1) students be used to obtain meaningful information and knowledge, from which a better understanding of present and future generations of CS-1 students can be derived, and solutions found to some of the academic problems and challenges facing them*?

An overview of the results for each sub-question leans towards a positive response to the main research question. Most of the factors show a significant influence on the academic performance of the `CS-1` students in the `COMS1000` unit. The results found are meaningful and can assist in the understanding of the present `CS-1` students. Moreover, since most of the research sub-questions involved the predicting of the academic performance of future generations, this research would be useful in assisting both current and future `CS-1` students. The research sub-questions that focus on "the module perceived to be the most difficult" and "selection of courses" will also assist with challenges faced by `CS-1` students. The supporting evidence is presented in the results of the six research sub-questions.

The sub-questions focus on the different factors in the Computer Science students social identity and academic life. The research focuses on the influence that the following five specific aspects have on academic performance of the Computer Science I students: Race and Gender, 'module perceived as most difficult', Mid-year performance, Mathematics I results, and Course combination. The final aspect is that of the influence of Computer Science I itself on the students overall first-year academic performance. These factors hint at a link between academic performance and social identity, course curriculum and/or degree factors. That gives a holistic picture of the first-year Computer Science student. This research is beneficial to both the academic stake holders (students, lecturers, course coordinators, administrators and the institution) and the Educational Data Mining community. This chapter will focus on each of the six secondary questions raised, the results presented and the association between the results with what was expected and related literature.

Section 4.2 presents the results for the investigation of the relationship between race, gender and academic performance in Computer Science I. No significant relationship was found between these factors and it can be concluded that on their own race and gender cannot conclusively be clustered according to

academic performance. In South Africa, a country with a history of demographic discrimination, it was important to analyse the influence of both these factors on success in education. The investigation of this particular question was very relevant in a society where woman were stereotyped as being better with household issues instead of technology. Black women, in particular, have historically faced restricted options of educational programmes and social roles because they have carried the responsibilities of family, social, and community development [Fisher *et al.* 1997; Clegg and Trayhurn 2000; Wilson *et al.* 2004; Mama 2003]. Numerous studies have also shown that the negative stereotypes about young women's abilities in mathematics, lower their test performance and aspirations for STEM (Science, Technology, Engineering and Mathematics) careers. Although the results were inconclusive, there are some notes that can be taken from the investigation. An initial look at the data showed that a majority of the `CS-1` students in terms of race were black and in terms of gender are male. It is expected that there would be fewer females in the `COMS1000` class. Roberts *et al.* [2002] is only one of the many articles that focus on under representation of females in Computer Science while Lagesen [2007] looks at strategies to increase the females in Computer Science. Results from the clustering of the data using the four different algorithms (EM, Farthest first, Simple K-means and MDBC) highlighted some interesting outcomes and cluster analysis as a potential data mining technique to explore further.

Focusing on race first, the cluster with the higher performing students consisted mainly of white students. The black students were distributed amongst the four other clusters with the majority of them being in the cluster of students that don't perform very well. A study in the States, by Katz *et al.* [2003], conducted a programming tutorial on students of different race and gender. Although Katz *et al.* [2003] mentioned a minority of African Americans in the study, they also found that the African American students did not perform as well as the White and Asian students in the tutorial. Although this study is in a different part of the world with a different society, it is an interesting finding which matches with what this research found. Looking at the gender results, it is clear that there are more males in the `COMS1000` class and more males are unsuccessful in their academic performance. Katz *et al.* [2003] shows that both males and females performed better than each other in different parts of the tutorial and could not conclude on which gender is more successful academically. The results from the gender part of the investigation show that more males are failing `COMS1000` than females. However it is also very important to note that the high achievers in the class are mostly male. A study by Siraj and Abdoulha [2009] is in line with these results stating that more females than males complete their studies in Science, Medicine and Art successfully.

The curriculum of a course is important when assessing factors that contribute to the academic performance of students in any course. In this research two factors from the Computer Science I curriculum were investigated: the module perceived as being the most difficult in the `COMS1000` course and the academic performance in the middle of the year (first semester). Results from both these investigations are presented in Section 4.3 and 4.4 respectively. Results from Section 4.3 show that in this instance, the module which is perceived to be the most difficult in the course has a correlation with the academic performance in the course. The LoC mark can be used to predict Computer Science I performance with minimal incorrect predictions. This is in agreement with the expected results as a directly proportional relationship was expected between performance in LoC and the `COMS1000` unit. This also brings a new factor for the EDM community to investigate since literature shows no research in this regard. It would be interesting to find a criteria for judging the most difficult course, which should be a mixture of student and lecturer opinions and facts of academic performance in that course.

The second curriculum factor is mid-year performance and its influence on final year-end `COMS1000` academic performance. Section 4.4 presented the results from the modelling of a dataset consisting of the mid-year `CS-1` marks and final year-end `CS-1` grades using three classification algorithms. The classification performed by the J48 Classifier in the SOFD tool predicted 92.54% of the final `CS-1` results correctly. This is a significant number and encourages not only the use of this tool but also the investi-

gation of the mid-year result in other courses. This result differs from the analysis in the O'Byrne *et al.* [2009] paper where it was found that the marks from the first half of the semester cannot conclusively predict students at risk of failing that semester. O'Byrne *et al.* [2009] did however use the in-semester mark rather than the final end of semester mark, which this research used. The use of the mid-year/ end of semester mark will require further investigation using a larger dataset and possibly other courses in other schools within the University of the Witwatersrand. This is still a significant result which is a contribution to the EDM community. It was expected that a relationship would be found between performance in the mid year assessments and the final year results. Additionally, as expected this prediction of the final year-end academic performance will enable the early intervention which could assist students who were found to be at risk of failing. With a result as good as 92.54% majority of the 'at risk' students would have been identified by this tool. It is noteworthy that analysis of some of the factors that were used to predict performance in Computer Science I, specifically the "module perceived to be the most difficult" and the mid-year marks, could have been done using regression analysis. However a major part of this research was to explore techniques being used in literature in the educational data mining sphere. For this reason the classification algorithms were explored.

When considering the curriculum of the `CS-1` students, it is also important to note that they do not only register for Computer Science I. The `CS-1` students are also required to register for, on average, three additional courses including the compulsory co-requisite Mathematics I. This is noteworthy because these additional courses could potentially impact their academic performance in the `CS-1` course and in the degree as a whole. The additional courses could affect performance in `CS-1` in multiple ways, such as knowledge of the course could be important to do well in Computer Science or the workload in that other course could be too much for balancing with the `CS-1` workload. In this research, the Mathematics I and Computer Science I relationship is first investigated, thereafter the Computer Science I relationship with the other courses was looked into.

The results from the Mathematics investigation showed a moderately strong and positive correlation between Mathematics I and Computer Science I. It was accepted that these two courses are statistically significantly related to each other. Further analysis using linear regression presented a strong relation between the two courses, also highlighting that the `COMS1000` mark can be predicted using the `MATH 1` mark with a high accuracy. It was important to also look at the modules within each course and their relation with the courses and each other. And although there is a correlation between Computer Science I and both the Mathematics I modules, it is the relationship with Algebra that was found to be significant. The matching exploration with regards to Mathematics I and the Computer Science I modules showed results indicating a significant relationship between `MATH 1` and the Fundamental Algorithmic Concepts (FAC) module. Moreover, both Mathematics I modules (Algebra and Calculus) had a stronger relationship with FAC than any of the other Computer Science I modules. The Mathematics and Computer Science relationship has been investigated to great lengths in previous work. Mathematics has been connected with Computer Science, Computer Studies, Information technology and Programming [White and Sivitanides 2003; Eid and Millham 2012]. Therefore from related literature it was expected that there would be some relationship between Mathematics and Computer Science. Paulson [2002] goes on to state that Mathematics has always been linked to Computer Science. Looking more specifically at the relationship between Computer Science and the Mathematics modules, in the Eid and Millham [2012] study, there was a strong correlation between academic performance in `Calculus` and any IT related course. This is different from the results presented in this research, however Eid and Millham [2012] did not investigate the affiliation between `Algebra` and IT courses. This result therefore needs further investigation with specific focus. Mathematics was also found to influence academic performance in Fundamental Algorithmic Concepts (`FAC`) more than all other modules of Computer Science I. Since `FAC` includes topics such as graph theory and proofs which are present in Mathematics, this result is expected. Previous research has also shown the strong relationship between Mathematics and algorithmic or programming courses [Eid and Millham 2012]. Overall the results of the relationship between

Mathematics and Computer Science were expected while highlighting a few details which could have been overlooked in the past.

Over and above Mathematics I, students are required to register for at least two other first year courses. These courses can be from any school within the University. Appendix J.1 shows the list of courses from different schools that CS−1 students from 2006 to 2012 registered for. Some courses have more Computer Science I students than others. Courses such as Computational and Applied Mathematics, which is a recommended additional course, has more CS−1 students than any other course. While a course like "Film ,Visual and Performing Arts IA" only has less than five students. From the 39 courses that were selected by the CS−1 students, it is clear that the COMS1000 students have a large variety of interests. It is also important to take into consideration that courses from schools in other faculties and those which are not necessarily "connected" to Computer Science will have their own workload and require a bit more from the CS−1 student. This leads to the focus of this research question which is to identify the best course combination for the BSc. students who select Computer Science I as a major.

The first analysis of this dataset was done by looking at the most common course combinations amongst the CS−1 students. The two most common course combinations included Computational and Applied Mathematics with Physics/Economics. The Computational and Applied Mathematics inclusion is expected but the interesting inclusion is the Physics and Economics. Fornés *et al.* [2008] found that students who pass Programming also do well in Physics. In this research study it was found that Physics was the most selected course after CAM however not part of the best rules. This meant that Physics was not necessarily a good selection. Additionally, a comparison of enrolments in different courses taken by Computer Science I students was done. It is clear that the majority of the CS−1 students select Computational and Applied Mathematics (CAM) as an additional course (Chapter 4.6). This is expected with CAM being a recommended co-requisite of Computer Science and the courses being inter connected. The next most enrolled course is Economics. This is an interesting finding and further investigation into why most CS−1 students select this course will need to be conducted.

The rules showed a good academic performance for CS−1 students who selected the Biological and Chemistry stream. This is an interesting result, the insights from which may benefit other areas of research, such as bioinformatics [Nilges and Linge 2013]. This would encourage students selecting Computer Science I, Mathematics I and Biological or Chemistry courses. It is also still valid to consider Computational and Applied Mathematics as a recommended course since it has shown to have a good relationship with COMS1000 and academic success. Psychology courses also seem to appear in both phases of this research question investigation. This is an interesting combination that is not popular. This is a great method for assisting students to select courses that would contribute towards success when paired with COMS1000 and Math I. This will increase chances of COMS1000 students being successful academically.

Finally, the last sub-question proved to be difficult to answer with the overall poor accuracy in performance of all classifiers applied to the data. Even in the test phase all of the algorithms incorrectly predicted more than half of the final first-year progression outcomes. This means that the model is not a good one and the probability of the COMS1000 results being a factor for predicting overall first-year academic performance of CS−1 students is highly unlikely. Further investigation of this can be performed using different classification techniques and possibly different datasets. However from this investigation, no clear relationship has been found between COMS1000 academic performance and overall first-year academic performance. The result of the COMS1000 mark not being able to predict the final first-year progression outcome was not the expected result. No related literature in this area can be consulted.

Three data mining techniques were used in this research, namely classification, clustering and association rule mining. In the first two techniques, different classification and clustering algorithms were

compared while only the predictive Apriori algorithm was used to obtain association rules. For the first two techniques, algorithms were compared while only the predictive Apriori algorithm was used for Association Rule mining. Research question 2,3 and 6 all use the classification technique with the following algorithms: the J48 Classifier, Decision Table and Naïve Bayes. At some point of the research, each classifier outperformed the other classifiers depending on the dataset. The J48 Classifier was overall a better performer when using educational datasets. It is also important to note that the J48 Classifier and Decision Table seem to perform similarly in most of the investigations that were done for this research. It would be interesting to investigate further using different datasets, data sizes and attribute numbers. This is in line with related literature on the prediction of student performance using classification techniques. Yadav and Pal [2012] concludes that decision trees algorithms are effective when using previous years data to predict future academic performance. Little literature in educational data mining uses the Decision Table as a predictive algorithm for academic performance. The good performance in this instance should increase the use of this classifier in Educational Data Mining research. The second technique looked at comparing three of the four clustering algorithms used: Simple K-means, Expectation-Maximization (EM) and Farthest first algorithm. The final investigation using the Race, Gender and Academic performance in `COMS1000` data set focused on varying the size of the clusters to compare the clustering algorithms. Using four different K values (number of clusters) showed the Farthest first algorithm as being better than the EM and Simple K-means algorithm. It consistently had the lowest incorrectly clustered instances and therefore the lowest average incorrectly clustered instances. This is different to what was shown in the paper by Jain and Gajbhiye [2012] where the Simple K-means algorithm was found to be the better performing algorithm. However none of the three clustering algorithms clustered a great majority of the records correctly. These results are not conclusive and further investigation in this field should be done. In the next paragraph, the factors that were considered to have performed well in the research are summarized.

It was not surprising, when considering the literature, that Mathematics and Computer Science were found to have a strong relationship. It was however interesting to find that Algebra had a stronger influence on Computer Science I in this dataset than Calculus. The midyear mark is also seen as a good predictive factor enlightened by this research. This encourages further investigation into midyear marks from other courses being used as predictors of overall performance. What has also not been done previously, is looking at a module which is considered to be difficult and using that as a predictor of the overall course. That also showed to be a useful investigation. Overall the research showed great possibilities in predicting academic performance of Computer Science I students. Additionally paved the way in terms of verifying data mining techniques when using educational setting data. The SOFD tool which is a product of this research can be used for further research purposes and "student at risk" analysis within the school of Computer Science at Wits University. Opening this tool up for research will also allow other schools and faculties at Wits University to explore possibilities of predicting their own students.

Similarly to most other projects there were limitations when conducting this research study. One of the biggest limitations of this research was that the data that was collected came from different silos. That is an issue because it makes it harder to consolidate the datasets. Most of the consolidation was done in the pre-processing phase of each of the research sub-question investigations, however even the counts were not always the same. The datasets coming directly from the School of Computer Science had a more updated list with only students who had results against their names. Datasets from the University database had all students who had registered for Computer Science I in the beginning of the year regardless of cancellations or any other changes to their registration. Another limitation, specific to the Mathematics research sub-question, was the change in curriculum. From 2009, Mathematics marks were separated into an Algebra mark and a Calculus mark. This was different from previous years where only one mark was awarded for the entire Mathematics course. After consultation with the School of Mathematics, a decision was taken to use an average of the two results. Another limitation was due to the use of three different data mining techniques, there was no clear focus on any of the three, even though more research

questions used the classification techniques. This also meant that not a large number of the algorithms in each technique could be compared and used for the various datasets.

Chapter 6

Conclusion and Future Work

Across the country, there has been an increase in the number of students enrolling into higher education institutions in the past few years. This increase has also been noticeable at Wits University as a whole and more specifically in the first year Computer Science class in the Faculty of Science. When student numbers increase it is important to understand the students being taught, their behaviour and factors that influence their academic performance. The aim of this research is to use the area of Educational Data mining and its techniques to investigate data stored about first year Computer Science students with the purpose of finding patterns, trends and information that could assist with throughput and retention. Six research sub-questions were formulated to assist in responding to the main research question. The interesting findings in this research include the good relationship between Computer Science I performance and performance in three of the factors investigated: Mathematics I, mid-year mark and the "module perceived to be the most difficult in the course". On the surface Mathematics and Computer Science relation was expected, however further investigation highlighted strong correlations between Computer Science and Algebra and Mathematics and the Fundamental Algorithmic Concepts module. This result helps explain why Mathematics I is a compulsory co-requisite of Computer Science I at Wits University. The mid-year and "module perceived as most difficult" factors are two which have not been investigated at great lengths in data mining literature. Both these factors perform well in the prediction of the `COMS1000` academic performance is good for the School of Computer Science at Wits University since they will allow for identifying students who are at risk of failing. This could help in improving throughput and retention rates since students at risk of failing can get early intervention. Additionally the positive result observed when investigating these results should encourage an increase in investigations of these factors in other courses. The investigation of the association between first year course selection and overall academic performance also yielded noteworthy results. It is found that the courses a student selects to enrol in, concurrently with Computer Science I, do affect the student's overall academic performance. This result is important because this knowledge can help with the appropriate advising of students during the registration process.

All the findings from the different factors used in this study highlight an important fact, that information about the students and their academic performance can be obtained using data mining techniques. To summarize this research project, it can be said that data mining is useful in giving educational settings a better understanding of their students and the teaching and learning in their environments. Classification, Clustering and Association Rule Mining are only some of the data mining techniques. Further investigation with regards to this should be conducted. Additionally, there are a lot of different algorithms that have been developed in the different data mining techniques that need to be considered in the educational setting. The SOFD tool, which was created for this research, is also a good indicator for students at risk in the School of Computer Science at Wits University and will be a good tool for administrators/lecturers or further investigation.

Various aspects of this research can be further investigated. Recommendations for future work were highlighted in Chapter 5. The following are some of the suggestions, following from this research, that can be implemented as part of future work. It would be very useful to check whether or not the SOFD tool could be modified to cater for the various schools in the University. All these schools would need different training datasets and that could cause a variation in algorithm performance. It would also be interesting to increase the use of this tool to other years of study, not just the first-year class. An important investigation that should be conducted is also a focused individual view of the different data mining tools when using educational datasets. Apart from race and gender, other factors such as home language can be investigated as part of research that seeks to obtain more information about Computer Science I students. Furthermore, even though previous academic history is a much studied factor, it could be included in our context, to obtain useful information and knowledge about the previous academic history of the Computer Science I students who attend our classes. These and other factors of interest would provide for a more comprehensive investigation of Computer Science I students, who form part of the teaching and learning environment that exists within the School of Computer Science.

References

[Abdullah *et al.* 2011] Zailani Abdullah, Tutut Herawan, Noraziah Ahmad, and Mustafa Mat Deris. Mining significant association rules from educational data using critical relative support approach. *Procedia-Social and Behavioral Sciences*, 28:97–101, 2011.

[Affendey *et al.* 2010] Lilly Suriani Affendey, IHM Paris, N Mustapha, Md Nasir Sulaiman, and Z Muda. Ranking of influencing factors in predicting students academic performance. *Information Technology Journal*, 9(4):832–837, 2010.

[Agrawal and Srikant 2000] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. *Sigmod Record*, pages 439–450, 2000.

[Aher and Lobo 2012] Sunita B Aher and LMRJ Lobo. Best combination of machine learning algorithms for course recommendation system in e-learning. *International Journal of Computer Applications*, 41(6):1–10, 2012.

[Anderson and Moore 2009] Ross Anderson and Tyler Moore. Information security: where computer science, economics and psychology meet. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1898):2717–2727, 2009.

[Arentze *et al.* 2008] Theo Arentze, Henri Achten, and Bauke de Vries. *Decision Support Systems for Design and Planning*, 2008. Retrieved 2013/06/08, from `http://www.ds.arch.tue.nl/education/courses/7m840/Default.menu?frame=page`

[Baker and others 2010] RSJD Baker et al. Data mining for education. *International encyclopedia of education*, 7:112–118, 2010.

[Baker and Yacef 2009] Ryan SJD Baker and Kalina Yacef. The state of educational data mining in 2009: A review and future visions. *JEDM-Journal of Educational Data Mining*, 1(1):3–17, 2009.

[Bayer *et al.* 2012] Jaroslav Bayer, Hana Bydzovská, Jan Géryk, Tomás Obsivac, and Lubomir Popelinsky. Predicting drop-out from social behaviour of students. *Proceedings of the 5th International Conference on Educational Data Mining*, pages 103–109, 2012.

[Bhullar and Kaur 2012] Munpreet Singh Bhullar and Amritpal Kaur. Use of data mining in education sector. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1, pages 24–26, 2012.

[Bouckaert *et al.* 2013] Remco R Bouckaert, Eibe Frank, Mark Hall, Richard Kirkby, Peter Reutemann, Alex Seewald, and David Scuse. *WEKA Manual for Version 3-7-8*, 2013.

[Builders 2012] Information Builders. *Predictive Analytics for Business Intelligence*, 2012. Retrieved June 2013, from `http://www.infobuild.co.za/predictive-analytics-for-business-intelligence/`

[Butterfield and Crews 2012] Jeff Butterfield and Thad Crews. Casting a wider net: A longitudinal study exploring gender differences, influences and attitudes academic major selection in computing. *Computer and Information Science*, 5(2):2–10, 2012.

[Campbell and McCabe 1984] Patricia F Campbell and George P McCabe. Predicting the success of freshmen in a computer science major. *Communications of the ACM*, 27(11):1108–1113, 1984.

[Chan *et al.* 2008] AYK Chan, KO Chow, and KS Cheung. Online course refinement through association rule mining. *Journal of Educational Technology Systems*, 36(4):2007–2008, 2008.

[Chang 2006] Lin Chang. Applying data mining to predict college admissions yield: A case study. *New Directions for Institutional Research*, 2006(131):53–68, 2006.

[Chapman *et al.* 2000] Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rudiger Wirth. *CRISP-DM 1.0 Step-by-step data mining guide*, 2000.

[Cheewaprakokbit 2013] Pimpa. Cheewaprakokbit. Study of factors affecting academic achievement of undergraduate students in international program. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, March, 13-15 2013.

[Cheng and Greiner 1999] Jie Cheng and Russell Greiner. Comparing bayesian network classifiers. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 101–108. Morgan Kaufmann Publishers Inc., 1999.

[Clegg and Trayhurn 2000] Sue Clegg and Deborah Trayhurn. Gender and computing: Not the same old problem. *British educational research journal*, 26(1):75–89, 2000.

[Daley 2010] Frank Daley. Why college students drop out and what we do about it. *College Quartely*, 13(3), 2010.

[Dancey and Reidy 2004] Christine P. Dancey and John Reidy. *Statistics without Maths for Psychology: Using SPSS for Windows*. Prentice-Hall Inc, 2004.

[Dekker *et al.* 2009] Gerben W Dekker, Mykola Pechenizkiy, and Jan M Vleeshouwers. Predicting students drop out: A case study. *International Working Group on Educational Data Mining*, 2009.

[Du 2010] Hongbo Du. *Data mining techniques and applications an introduction*. Course Technology Cengage Learning, 2010.

[Durant and Smith 2004] Kathleen T Durant and Michael D Smith. Predicting unix commands using decision tables and decision trees. In *Proceedings of the Third International Conference on Data Mining*, pages 427–436, 2004.

[Eid and Millham 2012] Chaker Eid and Richard Millham. Should mathematics be a mandatory fundamental component of any it discipline? *American Journal of Business Education (AJBE)*, 6(1):67–72, 2012.

[Fisher *et al.* 1997] Allan Fisher, Jane Margolis, and Faye Miller. Undergraduate women in computer science: experience, motivation and culture. In *ACM SIGCSE Bulletin*, pages 106–110. ACM, 1997.

[Fornés *et al.* 2008] Arturo Fornés, José A Conejero, Antonio Molina, Antonio Pérez, Eduardo Vendrell, Andrés Terrasa, and Emilio Sanchis. Predicting success in the computer science degree using roc analysis. In *FECS*, pages 86–90, 2008.

[Friedman *et al.* 1997] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163, 1997.

[Gabriel 2008] Kathleen Faye Gabriel. *Teaching unprepared students: Strategies for promoting success and retention in higher education*. Stylus Publishing, LLC., 2008.

[Gal-Ezer *et al.* 2004] Judith Gal-Ezer, Tamar Vilner, and Ela Zur. Teaching algorithm efficiency at cs1 level: A different approach. *Computer Science Education*, 14(3):235–248, 2004.

[García *et al.* 2007] Enrique García, Cristóbal Romero, Sebastián Ventura, and Toon Calders. Drawbacks and solutions of applying association rule mining in learning management systems. In *Proceedings of the International Workshop on Applying Data Mining in e-Learning (ADML 2007), Crete, Greece*, pages 13–22, 2007.

[Ghani and Soares 2006] Rayid Ghani and Carlos Soares. Data mining for business applications: Kdd-2006 workshop. *ACM SIGKDD Explorations Newsletter*, 8(2):79–81, 2006.

[Hall *et al.* 2009] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[Hochbaum and Shmoys 1985] Dorit S Hochbaum and David B Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 10(2):180–184, 1985.

[Hoffman and Novak 1998] Donna L Hoffman and Thomas P Novak. Bridging the digital divide: The impact of race on computer access and internet use. 1998.

[Howles 2007] Trudy Howles. Preliminary results of a longitudinal study of computer science student trends, behaviors and preferences. *Journal of Computing Sciences in Colleges*, 22(6):18–27, 2007.

[Jackson *et al.* 2011] Linda A Jackson, Alexander von Eye, Edward A Witt, Yong Zhao, and Fitzgerald Hiram E. A longitudinal study of the effects of internet use and videogame playing on academic performance and the roles of gender, race and income in these relationships. *Computers in Human Behavior*, 27:228–239, 2011.

[Jain and Gajbhiye 2012] S Jain and S Gajbhiye. A comparative performance analysis of clustering algorithms. *Int. J. Adv. Res. Comput. Sci. Softw. Eng*, 2:441–445, 2012.

[Jain *et al.* 1999] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[Jain *et al.* 2010] Sapna Jain, M Afshar Aalam, and MN Doja. K-means clustering using weka interface. In *Proceedings of the 4th National Conference; INDIACom-2010 Computing for Nation Development*, 2010.

[Jank *et al.* 2006] Wolfgang Jank, Galit Shmueli, and Shanshan Wang. Dynamic, real-time forecasting of online auctions via functional models. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 580–585. ACM, 2006.

[Jian *et al.* 2008] Hua-Li Jian, Frode Eika Sandnes, Yo-Ping Huang, Li Cai, and Kris MY Law. On students' strategy-preferences for managing difficult course work. *Education, IEEE Transactions on*, 51(2):157–165, 2008.

[Kabakchieva 2013] Dorina Kabakchieva. Predicting student performance by using data mining methods for classification. *Cybernetics and Information Technologies*, 13(1):61–72, 2013.

[Katz *et al.* 2003] Sandra Katz, John Aronis, David Allbritton, Christine Wilson, and Mary Lou Soffa. Gender and race in predicting achievement in computer science. *Technology and Society Magazine, IEEE*, 22(3):20–27, 2003.

[Kirkby 2008] Richard Kirkby. *Attribute-Relation File Format (ARFF)*, November 2008. Retrieved 08 February 2015, from `www.cs.waikato.ac.nz/ml/weka/arff.html`

[Kohavi *et al.* 2002] Ron Kohavi, Neal J Rothleder, and Evangelos Simoudis. Emerging trends in business analytics. *Communications of the ACM*, 45(8):45–48, 2002.

[Kohavi 1995] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, pages 1137–1145, 1995.

[Kovacic 2010] Zlatko Kovacic. Early prediction of student success: Mining students' enrolment data. In *Proceedings of Informing Science & IT Education Conference*, pages 647–665, 2010.

[Kumar and Chadha 2012] Varun Kumar and Anupama Chadha. Mining association rules in students assessment data. *International Journal of Computer Science Issues*, 9(5):211–216, 2012.

[Kumar and Rathee 2011] Varun Kumar and Nisha Rathee. Knowledge discovery from database using an integration of clustering and classification. *International Journal of Advanced Computer Science and Applications*, 2(3):29–32, 2011.

[Lagesen 2007] Vivian Anette Lagesen. The strength of numbers strategies to include women into computer science. *Social Studies of Science*, 37(1):67–92, 2007.

[Landis and Koch 1977] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

[Larose 2014] Daniel T Larose. *Discovering knowledge in data: an introduction to data mining*. John Wiley & Sons, 2014.

[Letseka and Maile 2008] Moeketsi Letseka and Simeon Maile. *High university drop-out rates: A threat to South Africa's future*. Human Sciences Research Council Pretoria, 2008.

[Lewis-Beck *et al.* 2003] Michael Lewis-Beck, Alan E Bryman, and Tim Futing Liao. *The Sage encyclopedia of social science research methods*. Sage Publications, 2003.

[Lima *et al.* 2009] Elen Lima, Christophe Mues, and Bart Baesens. Domain knowledge integration in data mining using decision tables: case studies in churn prediction. *Journal of the Operational Research Society*, 60(8):1096–1106, 2009.

[Lodhi *et al.* 2011] B. Lodhi, S. Jalori, and V. Namdeo. Result analysis by decision tree, decision table and rbf network. *Journal of Computer and Mathematical Sciences*, 2:399–580, June 2011.

[MacGregor 2007] Karen MacGregor. South africa: Student drop-out rates alarming. *University World News*, 2007.

[MacGregor 2009] Karen MacGregor. South africa: Shocking results from university tests. *University World News*, 16, 2009.

[Mama 2003] Amina Mama. Restore, reform but do not transform: The gender politics of higher education in africa. *Journal of Higher Education in Africa*, 1(1):101–125, 2003.

[Markov and Russell 2006] Zdravko Markov and Ingrid Russell. An introduction to the weka data mining system. *ACM SIGCSE Bulletin*, 38(3):367–368, 2006.

[Mashiloane and Mchunu 2013] Lebogang Mashiloane and Mike Mchunu. Mining for marks: A comparison of classification algorithms when predicting academic performance to identify students at risk. In *Mining Intelligence and Knowledge Exploration*, pages 541–552. Springer, 2013.

[Mashiloane 2014] Lebogang Mashiloane. Using association rule mining to find the effect of course selection on academic performance in computer science i. In *Mining Intelligence and Knowledge Exploration*, pages 323–332. Springer, 2014.

[Menard 2002] Scott Menard. *Longitudinal Research*, volume 76. Sage Publications, 2002.

[Merceron and Yacef 2005] Agathe Merceron and Kalina Yacef. Educational data mining: a case study. In *AIED*, pages 467–474, 2005.

[Mirkin 2012] Boris Mirkin. *Clustering: a data recovery approach*. CRC Press, 2012.

[Moja 2006] T Moja. Challenges in the internationalization of higher education in south africa. *The Internationalization of Higher Education in South Africa, Durban, South Africa: IEASA*, 2006.

[Mortagy and Boghikian-Whitby 2010] Yehia Mortagy and Seta Boghikian-Whitby. A longitudinal comparative study of student perceptions in online education. *Interdisciplinary Journal of E-Learning and Learning Objects*, 6(1):23–44, 2010.

[Myatt and Johnson 2009] Glenn J Myatt and Wayne P Johnson. *Making sense of data II: A practical guide to data visualization, advanced data mining methods, and applications*. John Wiley & Sons, 2009.

[Nel and Müller 2010] Norma Nel and Heléne Müller. The impact of teachers' limited english proficiency on english second language learners in south african schools. *South African Journal of Education*, 30(4):635–650, 2010.

[Nghe *et al.* 2007] Nguyen Thai Nghe, Paul Janecek, and Peter Haddawy. A comparative analysis of techniques for predicting academic performance. In *Frontiers In Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE'07. 37th Annual*, pages T2G–7. IEEE, 2007.

[Nilges and Linge 2013] Michael Nilges and Jens Linge. *Bioinformatics*, 2013. Retrieved 04 June 2013, from `http://www.pasteur.fr/recherche/unites/Binfs/definition/bioinformatics_definition.html`

[Nowozin 2012] Sebastian Nowozin. Improved information gain estimates for decision tree induction. *arXiv preprint arXiv:1206.4620*, 2012.

[O'Byrne *et al.* 2009] J.W. O'Byrne, S. Britton, A. George, S. Franklin, and A. Frey. Using academic predictors to identify first year science students at risk of failing. *CAL-laborate International*, 17, 2009.

[Osmanbegović and Suljić 2012] E. Osmanbegović and M. Suljić. Data mining approach for predicting student performance. *Economic Review*, 10(1), 2012.

[Pandey and Pal 2011] Umesh Kumar Pandey and Saurabh Pal. Data mining: A prediction of performer or underperformer using classification. *arXiv preprint arXiv:1104.4163*, 2011.

[Pasensie 2010] K. Pasensie. The end of obe? In *Southern African Catholic Bishops' Conference Briefing Paper 233*, 2010.

[Paulson 2002] Hasmik Gharibyan Paulson. Computer science students need adequate mathematical background. In *2nd International Conference on the Teaching of Mathematics (at the undergraduate level)*, July 2002.

[Pechenizkiy *et al.* 2008] Mykola Pechenizkiy, Seppo Puuronen, and Alexey Tsymbal. Does relevance matter to data mining research? In *Data Mining: Foundations and Practice*, pages 251–275. Springer, 2008.

[Penn-Edwards 2004] Sorrel Penn-Edwards. They do better than us!: A case study of course combinations and their impact on english assessment results. *Educating: Weaving Research into Practice*, 3, 2004.

[Pollock 2009] Steven J Pollock. Longitudinal study of student conceptual understanding in electricity and magnetism. *Physical Review Special Topics-Physics Education Research*, 5(2):020110, 2009.

[Rauchas *et al.* 2006] Sarah Rauchas, Benjamin Rosman, George Konidaris, and Ian Sanders. Language performance at high school and success in first year computer science. In *ACM SIGCSE Bulletin*, pages 398–402. ACM, 2006.

[Revathi and Nalini 2013] S Revathi and T Nalini. Performance comparison of various clustering algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(2):67–72, 2013.

[Rienks and Taylor 2009] Jane Rienks and Stephanie Taylor. Attrition and academic performance of students identified as at-risk using administrative data alone. In *First year in Higher education Conference*, 2009.

[Riesenfeld 2011] R. Riesenfeld. *Bayes' Theorem*, 2011. Retrieved 10 February 2013, from `http://www.eng.utah.edu/~cs5961/Resources/bayes.pdf`

[Roberts *et al.* 2002] Eric S Roberts, Marina Kassianidou, and Lilly Irani. Encouraging women in computer science. *ACM SIGCSE Bulletin*, 34(2):84–88, 2002.

[Romero *et al.* 2008] Cristóbal Romero, Sebastián Ventura, and Enrique García. Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 51(1):368–384, 2008.

[Salazar *et al.* 2004] A Salazar, J Gosalbez, I Bosch, R Miralles, and L Vergara. A case study of knowledge discovery on academic achievement, student desertion and student retention. In *Information Technology: Research and Education, 2004. ITRE 2004. 2nd International Conference on*, pages 150–154. IEEE, 2004.

[Sanders and Mueller 2000] Ian Sanders and Conrad Mueller. A fundamentals-based curriculum for first year computer science. *ACM SIGCSE Bulletin*, 32(1):227–231, 2000.

[Sanderson 2008] Robert Sanderson. *COMP527: Data Mining*, January 2008. Dept. of Computer Science, University of Liverpool.

[Schmah *et al.* 2010] Tanya Schmah, Grigori Yourganov, Richard S Zemel, Geoffrey E Hinton, Steven L Small, and Stephen C Strother. Comparing classification methods for longitudinal fmri studies. *Neural Computing*, 22:2729–2762, 2010.

[Shi and Conrad 2009] Runhua Shi and Steven A Conrad. Correlation and regression analysis. *Annals of Allergy, Asthma and Immunology*, 103(4, Supplement 1):S35 – S41, 2009.

[Siraj and Abdoulha 2009] Fadzilah Siraj and Mansour Ali Abdoulha. Uncovering hidden information within university's student enrollment data using data mining. In *Modelling & Simulation, 2009. AMS'09. Third Asia International Conference on*, pages 413–418. IEEE, 2009.

[Spiegel *et al.* 2009] Murray R. Spiegel, John Schiller, and Alu R. Srinivasin. *Probability and Statistics*. McGraw-Hill, 2009.

[Steyn *et al.* 2014] M Steyn, Teresa Harris, and C G Hartell. Institutional factors that affect black south african students' perceptions of early childhood teacher education. *South African Journal of Education*, 34(3), 2014.

[Strydom *et al.* 2010] JF Strydom, M Mentz, and GD Kuh. Enhancing success in higher education by measuring student engagement in south africa. *Acta Academica*, 42(1):259–278, 2010.

[Sumithra and Paul 2010] R Sumithra and S Paul. Using distributed apriori association rule and classical apriori mining algorithms for grid based knowledge discovery. In *Computing Communication and Networking Technologies (ICCCNT), 2010 International Conference on*, pages 1–5. IEEE, 2010.

[Superby *et al.* 2006] Juan-Francisco Superby, JP Vandamme, and N Meskens. Determination of factors influencing the achievement of the first-year university students using data mining methods. In *Workshop on Educational Data Mining*, pages 37–44, 2006.

[Taylor 1990] Richard Taylor. Interpretation of the correlation coefficient: a basic review. *Journal of diagnostic medical sonography*, 6(1):35–39, 1990.

[Tlabela *et al.* 2007] Kholadi Tlabela, Joan Roodt, Andrew Paterson, and G Weir-Smith. *Mapping ICT access in South Africa*. HSRC Press, 2007.

[Treasury 2010] National Treasury. *BUDGET 2010 - a people's guide*, 2010. Retrieved 18 October 2011, from `http://www.treasury.gov.za/documents/national%20budget/2011/guides/Eng%20PGB%202011_.pdf`

[Ury 2005] Gary Ury. A longitudinal study comparing undergraduate student performance in tradional courses to the performance in online course delivery. *The Information Systems Education Journal*, 3(20), 2005.

[Vannozzi *et al.* 2004] Giuseppe Vannozzi, Ugo Della Croce, Antonina Starita, Francesco Benvenuti, and Aurelio Cappozzo. Knowledge discovery in databases of biomechanical variables: application to the sit to stand motor task. *Journal of neuroengineering and rehabilitation*, 1:7, 2004.

[Velickov and Solomatine 2000] Slavco Velickov and Dimitri Solomatine. Predictive data mining: practical examples. In *2nd Joint Workshop on Applied AI in Civil Engineering*, 2000.

[Viera and Garette 2005] AJ. Viera and JM. Garette. Understanding interobserver agreement: the kappa stastic. *Fam Med*, 2005.

[Visa *et al.* 2011] Sofia Visa, Brian Ramsay, Anca L Ralescu, and Esther van der Knaap. Confusion matrix-based feature selection. In *MAICS*, pages 120–127, 2011.

[Vranic *et al.* 2007] M Vranic, Damir Pintar, and Z Skocir. The use of data mining in education environment. In *Telecommunications, 2007. ConTel 2007. 9th International Conference on*, pages 243–250. IEEE, 2007.

[White and Sivitanides 2003] Garry White and Marcos Sivitanides. An empirical investigation of the relationship between success in mathematics and visual programming courses. *Journal of Information Systems Education*, 14(4):409–416, 2003.

[White *et al.* 2009] Laura J White, John W Coffey, and Eman M. El-Sheikh. A longitudinal study of the effectiveness of teaching introductory computer programming online. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 529–536, 2009.

[Wiedenbeck *et al.* 2004] Susan Wiedenbeck, Deborah Labelle, and Vennila NR Kain. Factors affecting course outcomes in introductory programming. In *16th Annual Workshop of the Psychology of Programming Interest Group*, pages 97–109, 2004.

[Willmott and Matsuura 2005] Cort J Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79, 2005.

[Wilson *et al.* 2004] Fiona Wilson, Deborah Marlino, and Jill Kickul. Our entrepreneurial future: Examining the diverse attitudes and motivations of teens across gender and ethnic identity. *Journal of Developmental Entrepreneurship*, 9(3):177–197, 2004.

[Wirth and Hipp 2000] Rüdiger Wirth and Jochen Hipp. Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, pages 29–39, 2000.

[Witten and Frank 2005] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[Yadav and Pal 2012] Surjeet Kumar Yadav and Saurabh Pal. Data mining: A prediction for performance improvement of engineering students using classification. *World of Computer Science and Information Technology Journal (WCSIT)*, 2(2):51–56, 2012.

[Yurdugül and Askar 2013] H Yurdugül and P Askar. Learning programming, problem solving and gender: A longitudinal study. In *2nd World Conference on Educational Technology and Researches - WCETR2012*, pages 605–610. Elsevier, 2013.

[Zaïane 1999] Osmar R Zaïane. *Chapter 1: Introduction to data mining*, 1999. Retrieved February 2013, from `www.exinfm.com/pdffiles/intro_dm.pdf`

[Zaki *et al.* 2002] Mohammed J Zaki, Jason TL Wang, and Hannu TT Toivonen. Biokdd 2002: recent advances in data mining for bioinformatics. *ACM SIGKDD Explorations Newsletter*, 4(2):112–114, 2002.

Appendix A

Mathematics and Performance in CS-I-training and test.arrf Files

```
@RELATION mathtrainingdataset


@ATTRIBUTE MathMark NUMERIC
@ATTRIBUTE ComsGrade {PAS, FAL, PDS,FSB, FAB, WDF, WDA}

@DATA
33,PAS
28,FAL
38,PAS
64,FAL
31,PAS
38,FAL
33,PAS
33,PAS
42,PAS
54,FAL
28,PAS
62,PAS
50,PAS
62,PAS
85,PDS
65,PAS
51,PAS
80,PAS
61,PAS
86,PDS
54,PAS
47,PAS
56,PAS
59,FAL
63,PAS
91,PDS
63,PAS
```

Figure A.1: Sample of Mathematics .arff file of the training dataset

```
1  @relation mathtestdataset
2
3  @attribute MathMark numeric
4  @attribute ComsGrade {PAS,FAL,PDS,FSB,FAB,WDF,WDA}
5
6  @data
7  45.5,?
8  0,?
9  53.5,?
10 38,?
11 42,?
12 33,?
13 44,?
14 47.5,?
15 46,?
16 38,?
17 40,?
18 45,?
19 40.5,?
20 44.5,?
21 48,?
22 44.5,?
23 48,?
24 43,?
25 41,?
```

Figure A.2: Sample of Mathematics .arff file of the test dataset

Appendix B

Race and Gender- training and test.arrf Files

```
1  @relation 2006-2011Gender&Race-COMS
2
3  @attribute Race {Black,White}
4  @attribute Gender {F,M}
5  @attribute 'COMS Grade' {PAS,FAL,PDS,FSB,FNR,FAB,FDF,WDF,WDA}
6
7  @data
8  Black,F,PAS
9  Black,F,FAL
10 Black,M,PAS
11 Black,F,PAS
12 Black,M,FAL
13 White,M,PAS
14 Black,F,FAL
15 Black,M,PAS
16 Black,M,PAS
17 Black,M,PAS
18 Black,M,PAS
19 White,M,PAS
20 Black,F,PDS
21 Black,F,FAL
22 White,M,PAS
23 Black,M,PAS
24 Black,M,PAS
25 Black,M,PAS
26 Black,M,PAS
```

Figure B.1: Sample of Race and Gender.arff file of the training dataset

```
@relation 2006-2011Gender&Race-COMS

@attribute Race {Black,White}
@attribute Gender {F,M}
@attribute 'COMS Grade' {PAS,FAL,PDS,FSB,FNR,FAB,FDF,WDF,WDA}

@data
White,M,?
Black,M,?
Black,M,?
Black,M,?
Black,M,?
Black,M,?
Black,F,?
Black,M,?
Black,M,?
Black,M,?
White,M,?
White,M,?
White,M,?
Black,M,?
White,M,?
Black,M,?
Black,M,?
Black,F,?
White,M,?
```

Figure B.2: Race and Gender .arff file of the test dataset

Appendix C

Difficult module in the course- training and test.arrf Files

```
@RELATION loctofinaltrainingdataset


@ATTRIBUTE LOCMARK NUMERIC
@ATTRIBUTE ComsGrade {pas, fal, pds, fsb, fab, wdf, wda}

@DATA
71.1,pas
70.27,pas
80.54,pas
75.44,pas
0,fal
82.23,pds
78.62,pds
77.06,pds
61.48,fal
64.86,pas
66.74,pas
69.21,pas
50.16,pas
71.59,pds
56.27,pas
69,pas
76.73,pas
```

Figure C.1: Sample of Difficult course .arff file of the training dataset

```
@RELATION loctofinaltrainingdataset


@ATTRIBUTE LOCMARK NUMERIC
@ATTRIBUTE ComsGrade {pas, fal, pds, fsb, fab, wdf, wda}

@DATA
55,?
58,?
70,?
75,?
64,?
79,?
69,?
59,?
61,?
44,?
61,?
55,?
27,?
89,?
72,?
```

Figure C.2: Sample of Difficult course .arff file of the test dataset

Appendix D

Mid year to final mark- training and test.arrf Files

```
@RELATION midtofinaltrainingdataset

@ATTRIBUTE MIDMARK NUMERIC
@ATTRIBUTE ComsGrade {PAS, FAL,PDS, FSB, WDF, WDA, FDF, FNR, FAB}

@DATA
75.5,PAS
48.5,PAS
75.5,PAS
69,PAS
27,FAL
79.5,PAS
74.5,PAS
40,FAL
42,PAS
54.5,PAS
57,PAS
49.5,PAS
75,PAS
51,FAL
71.5,PAS
57.5,PAS
70.5,PAS
```

Figure D.1: Sample of .arff file of the training dataset

```
@RELATION midtofinaltestdataset

@ATTRIBUTE COMSMARK NUMERIC
@ATTRIBUTE ComsGrade {PAS, FAL,PDS, FSB, WDF, WDA, FDF, FNR, FAB}

@DATA
0,?
10.5,?
12.5,?
26,?
32.5,?
33.5,?
28,?
20,?
35.5,?
45,?
33.5,?
27.5,?
40,?
35.5,?
51,?
66.5,?
43,?
```

Figure D.2: Sample of .arff file of the test dataset

Appendix E

Course Combination- training and test.arrf Files

```
@relation coursecombo2006-2011nolessthan20

@attribute Computational_and_Applied_Mathematics_I {yes, no}
@attribute Physics_I_(Major) {yes, no}
@attribute Introduction_to_Psychology_I {yes,no}
@attribute Physics_I_(Auxiliary) {yes,no}
@attribute Economics_I {yes,no}
@attribute Business_Accounting_I {yes,no}
@attribute Chemistry_I {yes,no}
@attribute Basic_Principles_of_Individual_&_Group_Psychology_I {yes,no}
@attribute Economics_IB {yes,no}
@attribute ProgressionOutcome {PCD,RET,MFL,NCD,MRNM,MFC,MFE,MCE,MBZ,Q,INP,MFO,MBP,MBR,CAN,MCC,MCA}

@data
no,no,no,no,yes,no,no,no,no,RET
yes,no,no,no,yes,no,no,no,no,RET
yes,no,no,no,yes,no,no,no,no,RET
no,no,yes,no,no,no,no,yes,no,RET
yes,no,no,no,no,no,no,no,no,RET
yes,no,no,no,yes,no,no,no,no,RET
yes,no,no,no,no,no,no,no,no,RET
yes,no,no,no,no,no,no,no,no,RET
yes,no,no,no,yes,no,no,no,no,RET
yes,no,no,no,yes,no,no,no,no,RET
yes,no,no,no,no,no,no,no,no,PCD
yes,no,no,no,no,no,no,no,no,PCD
no,no,no,no,yes,yes,no,no,no,PCD
no,no,no,yes,no,no,no,no,no,PCD
yes,no,yes,no,no,no,no,yes,no,PCD
yes,no,no,no,yes,no,no,no,no,PCD
```

Figure E.1: Sample of .arff file of the training dataset

Appendix F

CS-I predicting overall first-year success-training and test.arrf Files

```
@RELATION comstofinaltrainingdataset

@ATTRIBUTE COMSMARK NUMERIC
@ATTRIBUTE ComsGrade {PAS, FAL,PDS, FSB, WDF, WDA, FDF, FNR, FAB}
@ATTRIBUTE PROGRESSIONOUTCOME {PCD, RET, MFL, NCD, MRNM, MFC, MFE, MCE, MBZ, Q, INP, MFO, MBP, MBR

@DATA
52,PAS,RET
22,FAL,RET
54,PAS,RET
50,PAS,RET
48,FAL,RET
71,PAS,RET
42,FAL,RET
59,PAS,RET
54,PAS,RET
53,PAS,RET
57,PAS,RET
69,PAS,RET
75,PDS,RET
48,FAL,RET
67,PAS,RET
60,PAS,RET
62,PAS,RET
```

Figure F.1: Sample of COMS to final .arff file of the training dataset

```
@RELATION comstofinaltestdataset

@ATTRIBUTE COMSMARK NUMERIC
@ATTRIBUTE ComsGrade {PAS, FAL,PDS, FSB, WDF, WDA, FDF, FNR, FAB}
@ATTRIBUTE PROGRESSIONOUTCOME {PCD, RET, MFL, NCD, MRNM, MFC, MFE, MCE, MBZ, Q, INP, MFO, MBP, MBR, CAN, MCC, MCA}

@DATA
89,PAS,?
42,FAL,?
0,FAB,?
57,PAS,?
62,PAS,?
61,PAS,?
62,PAS,?
50,PAS,?
54,PAS,?
54,PAS,?
78,PAS,?
65,PAS,?
60,PAS,?
66,PAS,?
71,PAS,?
65,PAS,?
56,PAS,?
44,FAL,?
58,PAS,?
```

Figure F.2: Sample of COMS to final .arff file of the test dataset

Appendix G

SOFD Tool GUI

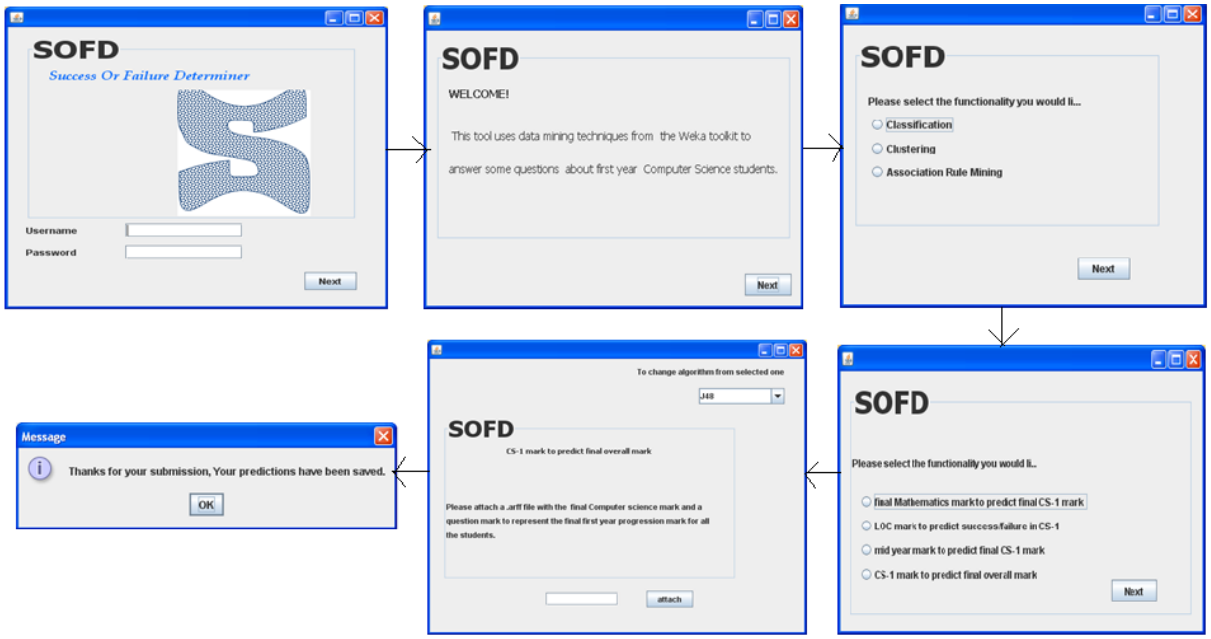


Figure G.1: Complete SOFD Tool

Appendix H

Individual SOFD GUIs

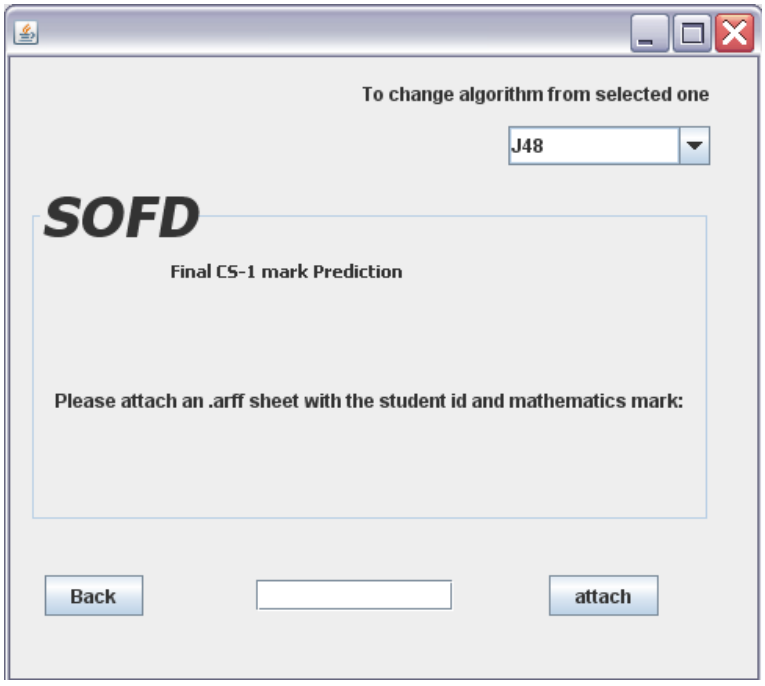


Figure H.1: SOFD 1

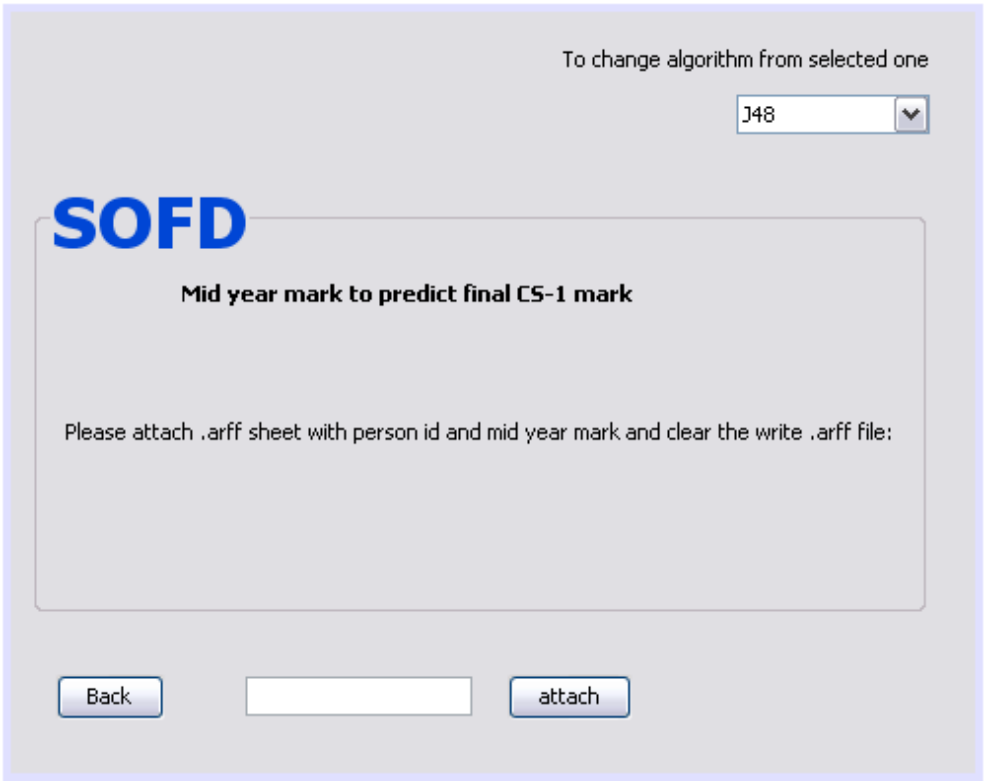


Figure H.2: SOFD 4

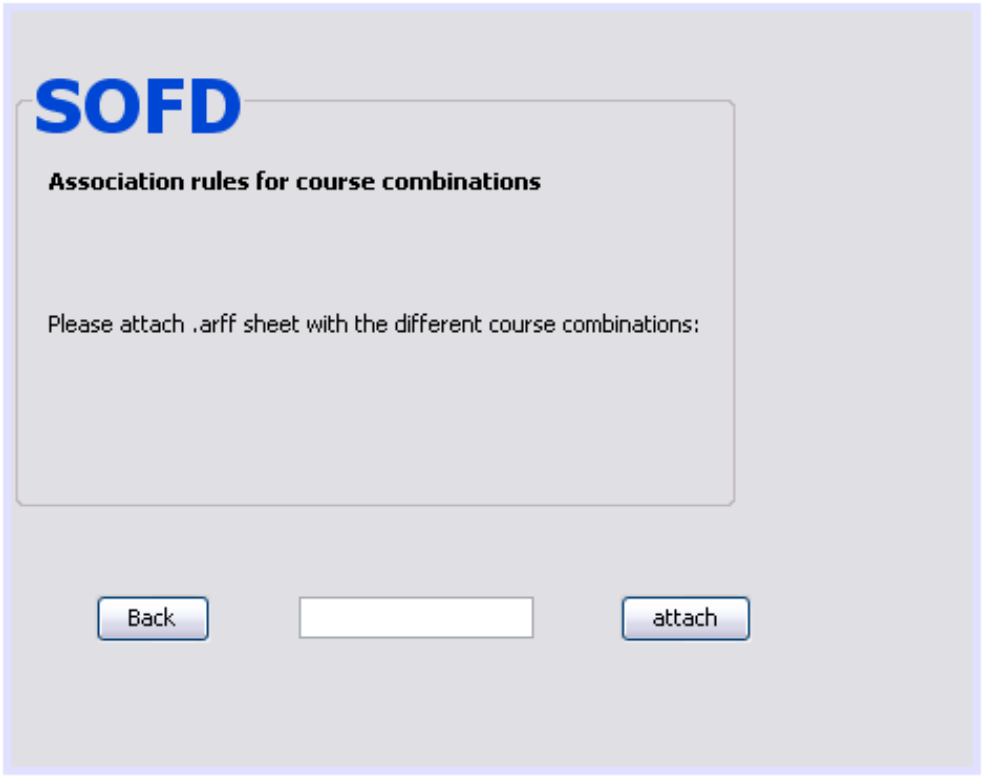


Figure H.3: SOFD 5

Appendix I

WEKA Knowledge Flow

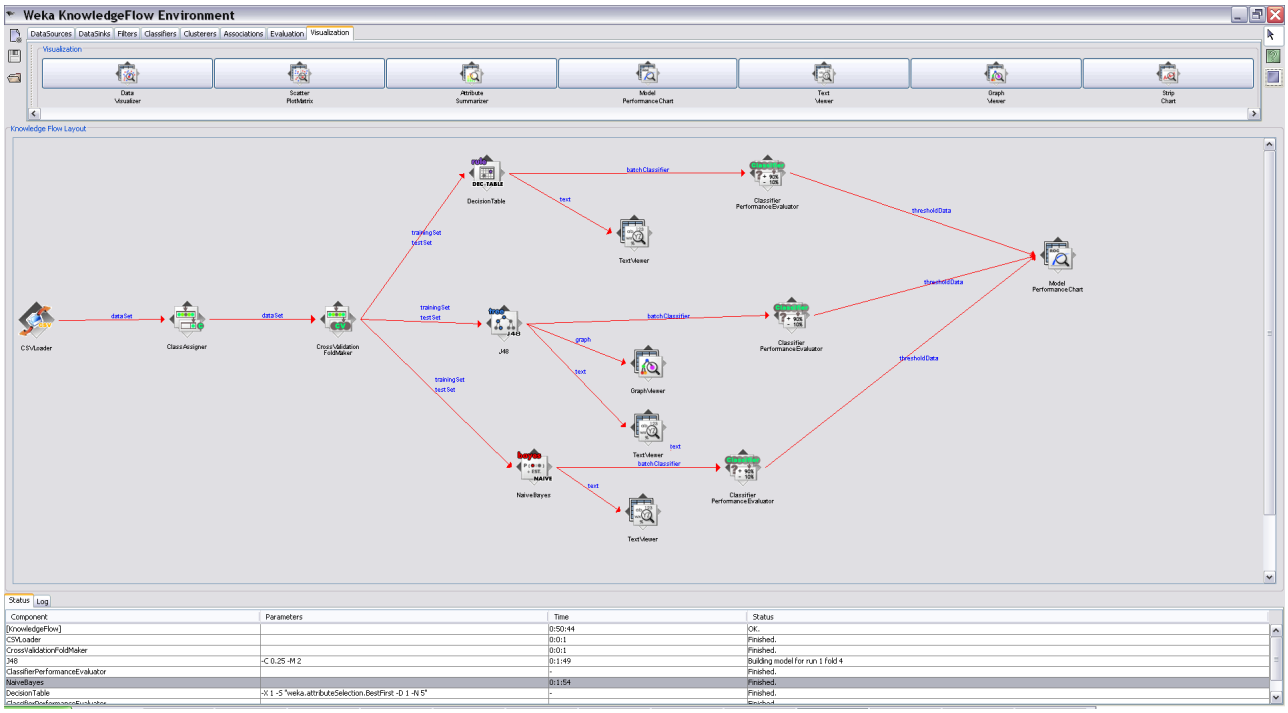


Figure I.1: WEKA Knowledge Flow for Mathematics Training set

Appendix J

List of Courses that COMS1000 students have registered for

```
Computational and Applied Mathematics I
Physics I (Major)
Introduction to Psychology I
Physics I (Auxiliary)
Geography I
Economics I
Business Accounting I
Chemistry I
Basic Principles of Individual & Group Psychology I
Mathematical Statistics I (HC)
Information Systems IA
Economics IIA
Critical Thinking & Philosophical Reasoning I
Economics IIB
Computational and Applied Mathematics for Applied Computing I
Physics for Applied Computing I
Information Systems IB
Economics IB
Concepts of Design
Economics IA
Archaeology I
Introduction to the Structure of Language I
Introductory Life Sciences
Basic German Language & Culture I
Chemistry IA
Chemistry IB
Accounting I
Actuarial Science I (HC)
Global Literature and Film I
Chemistry I (X,Y)
Language, Mind & Brain I
Introduction to Sociology A - Identity & Society I
Introduction to Sociology B - SA & Globalisation I
Psychological Research Design & Analysis IIA
Film, Visual and Performing Arts IA
Film, Visual and Performing Arts IB
Geology
English Literature in Context I
Personality and Developmental Psychology II
```

Figure J.1: List of Courses that COMS1000 students have registered for

Appendix K

Publication Efforts

1. A poster submitted and accepted for presentation at the 5th Cross-Faculty Graduate Symposium, held at Wits University, on 29 July - 02 August, 2013.
2. [Mashiloane and Mchunu 2013]
3. [Mashiloane 2014]