GUIDELINES FOR THE USE OF MACHINE LEARNING TO PREDICT STUDENT PROJECT GROUP ACADEMIC PERFORMANCE

R.V. EVEZARD

2020

Guidelines for the use of machine learning to predict student project group academic performance

By

Ryan Evezard

Submitted in fulfilment

of the requirements

for the degree of

Master of Information Technology

to be awarded at the

Nelson Mandela University

April 2020

Supervisor: Prof. Lynn Futcher

Co-supervisor: Prof. Johan van Niekerk

Declaration

I, Ryan Evezard, 213245035, hereby declare that the dissertation for Master of Information Technology is my own work and that it has not previously been submitted for assessment or completion of any postgraduate qualification to another University or for another qualification

Ryan Evezard

Official use:

In accordance with Rule G5.6.3,

5.6.3 A treatise/dissertation/thesis must be accompanied by a written declaration on the part of the candidate to the effect that it is his/her own work and that it has not previously been submitted for assessment to another University or for another qualification. However, material from publications by the candidate may be embodied in a treatise/dissertation/thesis

Abstract

Education plays a crucial role in the growth and development of a country. However, in South Africa, there is a limited capacity and an increasing demand of students seeking an education. In an attempt to address this demand, universities are pressured into accepting more students to increase their throughput. This pressure leads to educators having less time to give students individual attention.

This study aims to address this problem by demonstrating how machine learning can be used to predict student group academic performance so that educators may allocate more resources and attention to students and groups at risk. The study focused on data obtained from the third-year capstone project for the diploma in Information Technology at the Nelson Mandela University.

Learning analytics and educational data mining and their processes were discussed with an in-depth look at the machine learning techniques involved therein. Artificial neural networks, decision trees and naïve Bayes classifiers were proposed and motivated for prediction modelling. An experiment was performed resulting in proposed guidelines, which give insight and recommendations for the use of machine learning to predict student group academic performance.

Acknowledgements

I would like to thank my supervisors Prof. Lynn Futcher and Prof. Johan van Niekerk for their knowledge, patience and support throughout this research. Thank you for all your help and time. I would also like to thank the following benefactors for their financial assistance:

- The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the researcher, and are not necessarily to be attributed to the National Research Foundation.
- The financial assistance of the Nelson Mandela University's Post Graduate Research Scholarship (PGRS) is also hereby acknowledged.

Finally, I would like to thank my girlfriend, (Charne Keen) and my parents, (David and Michelle Evezard), who supported and motivated me through all my studies.

Table of Contents

List of Figures

List of Tables

List of Code Listings

Chapter 1: Introduction

1.1 Theoretical Framework

The growing student population has resulted in the loss of the individual student into the masses (Liu, Froissard, Richards, & Atif, 2015). Universities are specifically affected by this as the intake of students at higher education institutions is continually rising. Students are the main stakeholders of universities, which provide undergraduate and post-graduate education to their students. Universities are, in most ways, quite like profit-seeking organisations as they should achieve an operating surplus if they continue to provide the level of service expected of them (Adam & Nel, 2009). The main objective of higher education institutions in meeting this expectation, is to provide quality education to its students (Hamsa, Indiradevi, & Kizhakkethottam, 2016). These institutions and their ability to produce qualified graduates play an important role in the growth of a country (Yadav, Bharadwaj & Pal, 2012).

There are two aspects, which contribute to the number of graduates and, in turn, to the contribution of an institution to the development of a country. Firstly, student performance, which refers to the ability of students to complete their studies in the recommended number of years with good results. Secondly, the number of students that the university can service. The second aspect is especially true in South Africa where institutions are often pressured into accepting more and more students to increase their throughput rates (Boughey, 2003). The pressure of needing to accept more students leads to increased class sizes and usually means that educators have

less time to give students individual attention. This makes it nearly impossible to personalise the learning experience (Blatchford, Bassett, & Brown, 2011; Bersin, 2004).

In order to maintain high educational standards, universities must balance their allocation of resources between the need to provide quality education, which refers to each student getting the optimal span of personal attention, versus the need for the university to service as many students as possible (Van Niekerk & Webb, 2016). One way to provide quality education is to predict student academic performance and to use that information to improve the overall student performance (Hamsa, et al., 2016). Educators will then be able to use the predicted performance to identify which students need personalised attention. This helps universities to balance the number of students serviced while still being able to provide a personalised learning experience. In order to make improvements, students need feedback on what they have done and guidance on what they should do in the future. Predicting student academic performance will enable students to receive adequate feedback and guidance according to their specific needs (Kane, Kerr & Pianta, 2014). Romero and Ventura (2007) stated that the prediction of students' performance is an important aspect of higher education as the quality of teaching is attributed to the ability to meet students' needs.

There are two popular techniques used in student academic performance prediction, namely, educational data mining (EDM) and learning analytics (LA). These two techniques are closely tied to each other as they both use machine learning (Elias,

2011). Machine learning investigates how computer programs are able to learn and identify complex patterns and to make decisions based on data (Han, Kamber & Pei, 2011). When data mining is applied in education, it is referred to as EDM. EDM is one of the most popular techniques for predicting student academic performance (Hamsa, et al., 2016; Shahiri, Husain & Rashid, 2015). This method is used to extract useful information from an educational database. The information is then used to reveal applicable knowledge that was not immediately seen from the raw data (Romero & Ventura, 2007; Shahiri, et al., 2015). EDM uses various techniques of machine learning, including neural networks, decision trees and naïve Bayes classifiers in order to make predictions. Each of these techniques is discussed in detail in Chapter 3. By using these techniques, different knowledge types are revealed, such as classification, association rules and clustering (Baradwaj & Pal, 2012). In recent years, data mining in education has been receiving increased interest, as is evident in the review on EDM from 1995 to 2005 by Romero and Ventura (2007).

LA is the second most popular technique used to predict student academic performance. This field arose owing to the interest in how data from educational databases can be used to improve teaching and learning. Siemens and Long (2011, p. 34) define LA as "the measurement, collection, analysis, and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs". LA enables educators to predict and improve student success as it allows them to make decisions in education based on relevant data (Olmos & Corrin, 2012; Smith, Lange & Huston, 2012).

According to Van Harmelen and Workman (2012), LA and EDM can be used to:

- Identify students at risk of failing, in order to provide positive interventions;
- Analyse student academic performance, in order to provide personal recommendations related to learning materials and learning activities.

Baker's (2007) depiction of the knowledge continuum is represented in Figure 1.1. This is closely related to the process that LA and EDM follow to provide educators with the knowledge that enables them to make informed decisions. Data is the lowest part of the continuum and consists of raw input that is meaningless on its own. Data becomes information when it receives meaning. This information then becomes knowledge, through analysis and synthesis and finally, knowledge becomes wisdom, through the application of that knowledge.

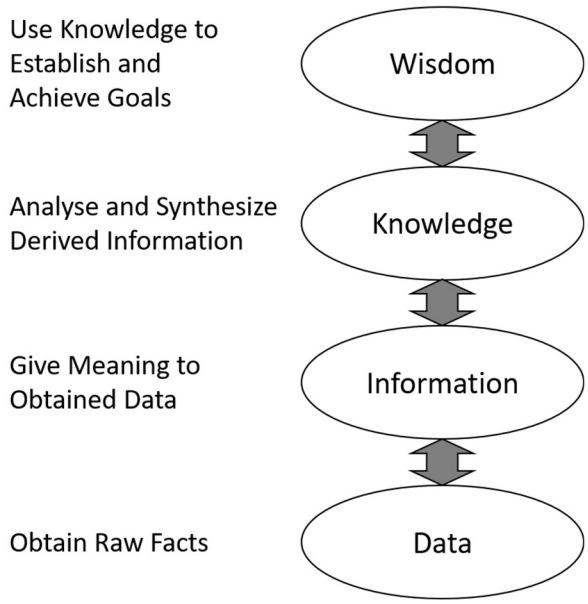

Figure 1.1: Depiction of the Knowledge Continuum (Baker, 2007)

The relationship between the knowledge continuum and LA and EDM becomes apparent when comparing it to the steps of extracting knowledge from data seen in Figure 1.2.

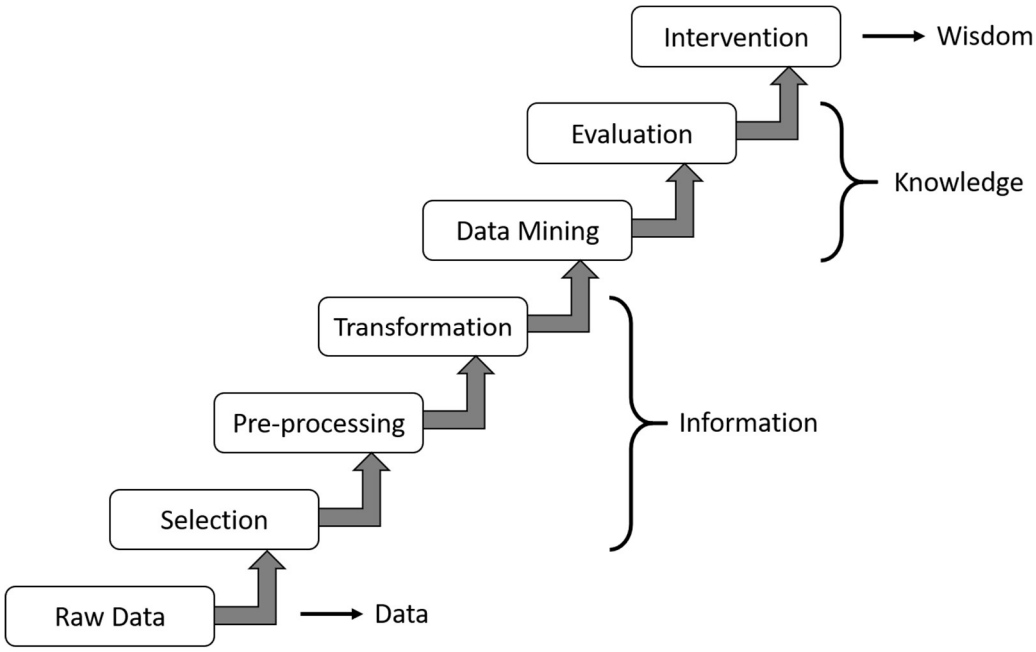

Figure 1.2: The Steps of Extracting Knowledge from Data (adapted from Baradwaj & Pal, 2012)

The data in Figure 1.1 refers to the same data as the raw data in Figure 1.2. Through selection, pre-processing and transformation, the data becomes information. Through data mining and evaluation that information then becomes knowledge. Lastly, wisdom is created through interacting with knowledge in some way. Educators would be able to classify students into groups based on their predicted academic performance, by using this wisdom. Applicable interventions can be applied based on each student's need for guidance (Romero & Ventura, 2007). It is apparent from Figures 1.1 and 1.2 that, in order to obtain knowledge, some form of raw data is initially required.

Students generate an enormous amount of data in the process of fulfilling their academic obligations. These obligations can include activities such as tests, assignments and practical assessments. This educational data may be used to identify students at risk early enough in the academic process, so that meaningful interventions can be made. EDM and LA can be used to obtain the hidden information contained in these large educational data repositories (Romero & Ventura, 2007). Daud, et al. (2017) state that a rich dataset containing multiple characteristics is required as the basis for these techniques.

Raw educational data is generated by several different sources. The first source of data is generated by students in performing academic activities. This includes all summative assessments, such as semester tests, assignments and practical assessments. The next source of data is a result of an increased popularity in e-learning. E-learning uses electronic technologies in order to distribute educational material to students (Fry, 2001). With the increased adoption of e-learning as a method for educators to share educational resources and material online, by 2010 institutions had started to use learning management systems (LMS) as a preferred method of distribution (Unwin, et al., 2010). A key aspect of an LMS is that it records all the interactions between a student and the system with varying levels of granularity depending on the LMS in question (Liu, et al., 2015). The LMS currently being used by the Nelson Mandela University is known as Moodle. Moodle is a free, online LMS enabling educators to create their own private website filled with dynamic courses that extend learning (Moodle, 2017). As a result of students interacting with Moodle, action logs are generated containing data relating

to their actions on Moodle. LMS, Moodle and the data generated are discussed in more detail in Chapter 2.

This data is being recorded and stored by the Nelson Mandela University. However, it is not currently being used to assist in making data-driven decisions nor in identifying students at risk and in need of some form of intervention. Currently, it is both difficult and resource-intensive for educators to be proactive and to identify which students are at risk of failing and needing further intervention. This, in turn, wastes more resources since students who could have passed with appropriate early intervention are currently failing, therefore requiring the time of educators and supervisors for an additional year. In contrast, being able to identify students at risk early on would reduce the amount of time that supervisors and educators need to spend on students who do not, in fact, need additional assistance.

In South Africa, there is currently a larger project called the Siyaphumelela Project, which is aimed at using student data in order to increase student success. As discussed in Chapter 2, there are also many academic publications that deal with the prediction of student performance as well as different analytics and student data that can be used to do so. However, these research projects generally focus on identifying *individuals* at risk, each of them using unique analytics. This research differs in that it focuses on students working in project groups, and it identifies whether an entire group might need an early intervention to prevent possible failure. One such context is students working in groups for third-year capstone projects towards Software Development qualifications at the Nelson Mandela University.

This module's weekly deliverables are managed by Moodle and interactions are recorded in the Moodle action logs. The project is divided into four iterations throughout the year, with feedback given at the end of each iteration. This, together with regular requirements to submit project documentation, generates a vast amount of student data; however, no effort has been made to analyse this data for educational decision-making purposes.

This lack of use of currently available data to make data-driven decisions and the identification of students at risk of failing, leads us to the problem to be addressed by this research.

1.2 Problem Statement

From the above literature, the problem statement of this research is defined as:

At the Nelson Mandela University, it is currently not known how educational data could be used for the early identification of third-year project groups that might be at risk of failure.

1.3 Research Objectives

The primary objective (PO) of this research in solving this problem is:

To provide guidelines for the use of machine learning techniques to predict academic performance of student project groups.

The following are secondary objectives (SO) that must be achieved for further development of the research to take place:

SO_1**:** *To identify common attributes of student activities used to predict academic performance.*

SO_2**:** *To identify what data and attributes are currently available in Moodle to be used for predicting student group academic performance.*

SO_3**:** *To determine the machine learning techniques commonly used by educational data mining and learning analytics and to argue their relevance to a specific data set.*

SO_4**:** *To use the selected machine learning techniques to create prediction models to predict student group academic performance.*

SO_5: *To evaluate and compare the performance of selected machine learning techniques in predicting student group academic performance.*

1.4 Delineation

The scope of this project is delineated to the ONT3660 module, which is the third-year capstone project module at the Nelson Mandela University. Assessment marks and Moodle action logs for this module alone from 2016 and 2017 were gathered. This data was then combined into a single data set and used as input data in order to predict the performance of project groups containing multiple students.

1.5 Research Design

The research methodology that will be followed during this research will be experimental methodology. Experimental methodology is used in research when one wants to prove some theory, test some theory or find something interesting. When experiments are used with the goal of finding something interesting it is known as exploratory (Olivier, 2009). This research will follow an exploratory experiment as no theory or proof has been established. This approach consists of designing the experiment, conducting the experiment, observing the results, theorising whether these results are interesting and finally, reporting on the results (Olivier, 2009). The experiment is designed and conducted in Chapter 4. Furthermore, the results of the experiment are observed and reported on in Chapters 5 and 6 respectively.

The goal of the experiment was to determine the effectiveness of using machine learning techniques to predict student group academic performance. The design process involved selecting relevant machine learning techniques and data. The

experiment was conducted by creating a prediction model based on the selected machine learning techniques. The focus was on incrementally determining the validity of the prediction models and modifying them constantly in order to optimise the accuracy measurement. Relevant metrics and analytic techniques were used to investigate to what extent the prediction models met the solution requirements (Von Alan, March, Park, & Ram, 2004).

1.6 Chapter Outline

The chapter outline of the study is shown in Table 1.1.

Table 1.1: Chapter Outline

Chapter 1	Introduction
Chapter 2	Learning Analytics and Educational Data Mining
Chapter 3	Machine Learning
Chapter 4	Experimental Design
Chapter 5	Prediction Model Analysis
Chapter 6	Proposed Guidelines
Chapter 7	Conclusion

1.7 Ethical Considerations

This project adheres to all requirements for research ethics as mandated by the Nelson Mandela University. No ethical clearance was required for this study.

1.8 Conclusion

This chapter served as an introduction to this study. It introduced learning analytics and educational data mining as well as their underlying processes. The problem to be addressed by this study was stated together with the objectives, which serve to address this problem. Finally, the scope, research design and chapter outline were discussed. From this chapter, it is evident that there is a lack of data-driven decision making for predicting student group academic performance at the Nelson Mandela University. Chapter 2 introduces learning management systems and provides a more in-depth understanding of learning analytics and educational data mining together with some problems associated with these techniques. Furthermore, Chapter 2 identifies common attributes used to predict student academic performance from related studies.

Chapter 2: Learning Analytics and Educational Data Mining

2.1 Introduction

The problem addressed by this study is stated in Chapter 1. This study warrants the need to conduct a literature review in order to analyse the problem within a real-world context. This chapter reports on a thorough review of existing literature in order to define learning analytics and educational data mining with regard to the processes and data involved therein. The layout and research objectives of the chapter are illustrated in Figure 2.1. The following research objectives (Section 1.3) are addressed in this chapter:

SO_1: *To identify common attributes of student activities used to predict academic performance.*

SO_2: *To identify what data and attributes are currently available in Moodle to be used for predicting student group academic performance.*

Before designing the prediction models, it is important to investigate learning management systems (LMS) and their ability to collect data (Section 2.2). Moodle will be investigated specifically owing to its relevance to the study. Learning analytics (LA) (Section 2.3) and educational data mining (EDM) (Section 2.4) provide an opportunity to bridge the gap between the LMS and the proposed

prediction models. It is imperative that this study identifies various issues with LA and EDM that may hinder the design of the prediction models (Section 2.5). There is a variety of data that can be collected from educational systems (Section 2.6). An investigation of existing research may give insight into useful attributes used to predict student academic performance.

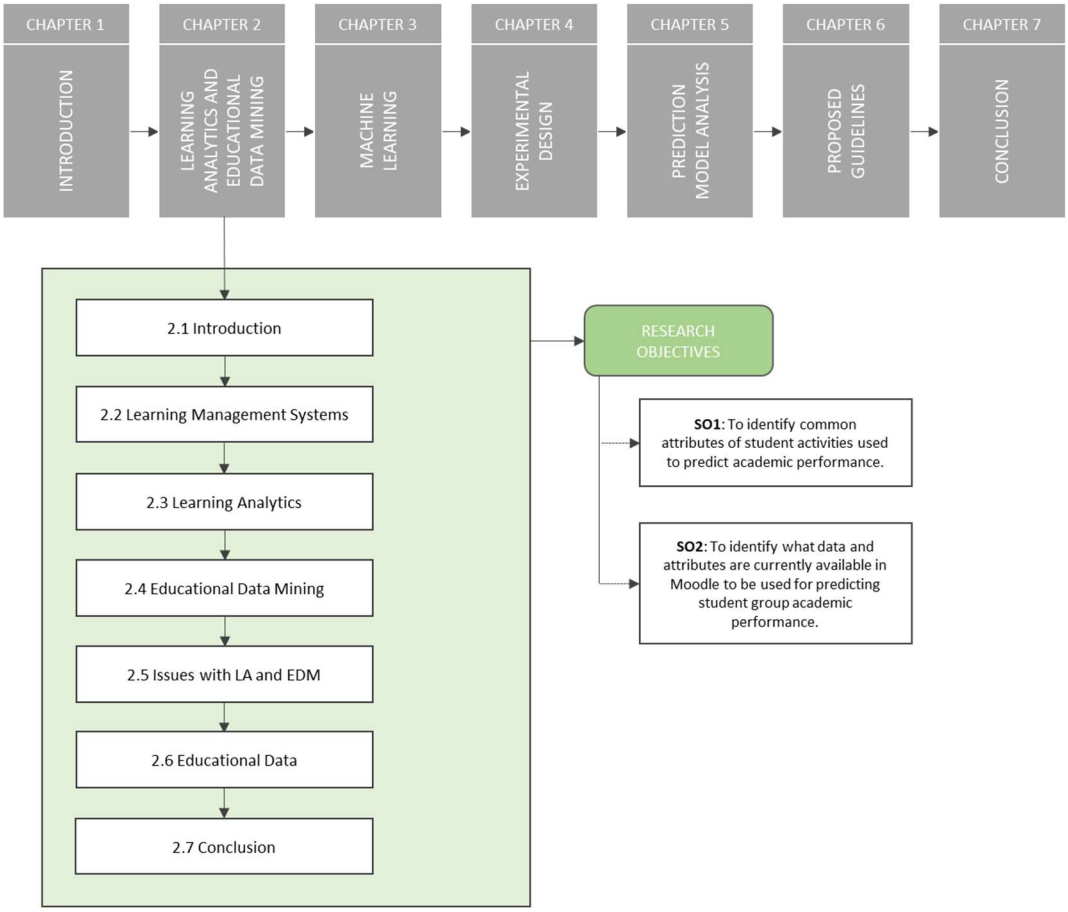

Figure 2.1: Chapter Two Outline

2.2 Learning Management Systems

LMS are web-based systems that facilitate online teaching and learning. These systems employ tools that provide interaction between students and educators. Using these tools enables educators to perform tasks such as sharing learning resources with students, creating online assessments and allowing students to upload documents (Unwin, et al., 2010). Institutions are adopting LMS, along with traditional face-to-face learning, in order to improve the quality of the holistic student learning experience. Institutions are able to use these systems to conduct distance- and blended learning, in order to meet the needs of the growing student population (Unwin, et al., 2010). During online- or blended learning, students interact and perform activities through the use of an LMS (Yu & Jo, 2014). It is an increasing trend for educators to use an LMS as part of a traditional course. This provides opportunities for both students and educators to increase learning engagement (Dahlstrom & Bichsel, 2014). The availability of course material that an LMS provides positively impacts the ability of students to learn inside and outside of the classroom. Evidence of improvement has been found both in student learning skills (Nair & Patil 2012), and in academic performance (Ebardo & Valderama, 2009).

SkyPrep (2017) identified six key features of an LMS, including course and content management, user management, reporting and analytics, white labelling, integration and security. These features are as follows:

1. The primary element and purpose of an LMS is *course and content management*. An LMS allows administrators to create and manage course and learning material effectively.
2. LMS should facilitate effective *user management*. This includes the automation of tasks such as forming student groups, managing registration, and deactivation of accounts.
3. *Reporting and analytic* tools are provided by LMS to allow for tracking and assessing student performance and progress.
4. *White labelling* is a feature of an LMS that allows the institution to customise the appearance of the LMS to match the institution's brand. The implication of this is that users of the customised LMS think they are using an internal tool.
5. For a company to synchronise data and information between an LMS and internal tools efficiently, the LMS should be able to *integrate* easily with other systems being used.
6. Finally, it is essential that LMS are *secure* to ensure that private user and organisational information can be stored safely in the system.

In LMS, the behaviour data of users are recorded in action logs (Macfadyen & Dawson, 2010). These systems accumulate large amounts of data on student behaviour that can be gathered and used to improve student interactions within an LMS (Beer, Clark, & Jones, 2010). This information includes users' visits, number of downloads, LMS tools accessed, messages read or posted, and content pages visited (Macfadyen & Dawson, 2010).

LMS can be grouped into two main categories: open source systems, such as Moodle, Sakai and Whiteboard; and proprietary solutions, such as WebCT, DesireLearn and Gradepoint. Open source means that the source code can be modified to suit the specific needs of the organisation (Cavus, Uzunboylu & Ibrahim, 2007). New features can be added, bugs can be fixed, and performance can be improved. Open source LMS are generally free of charge and based online (Pappas, 2017). By 2011, almost 80% of educational institutions worldwide had installed an LMS, with Moodle being the most popular (Munguatosha, Muyinda, Lubega, 2011). For the purpose of this study, and owing to Moodle's being the LMS used by the Nelson Mandela University in the case study, Moodle is discussed in detail. LMS such as Sakai provide similar functionality to Moodle. However, they have the added functionality of generating statistics. These statistics are presented in the form of summaries and custom support regarding user visits, tool use and course activity (Sakai, 2018). Currently, these statistics are not available in Moodle.

Moodle (modular object-oriented dynamic learning environment) is an open source web application that educators can use to construct effective online learning sites (Lopes, 2011). The system was originally developed by Martin Dougiamas in 1999, but because of its open source nature, it had grown in functionality by 2010. It has since grown even more with the latest version being Moodle 3.7.1, as of 2019.

The Moodle platform has three levels of use, with features of differential use and access, namely administrator; educator; and student. The administrator is the manager of the LMS and has full access and control. The educator has moderate

access and control but is able to perform actions such as creating activities. The student has limited access and is usually only able to access available materials or to participate in activities (Moodle, 2017).

Moodle has a set of activities available to its users. These activities are outlined and briefly described in Table 2.1.

Table 2.1: Moodle Activities (Moodle, 2017)

Activity	Description
Assignment	Allows students to upload file submissions and enables educators to mark and add comments to the submissions.
Chat	Allows students and educators to have conversations online.
Choice	Provides a means for an educator to gather responses from students for specified questions, similar to a multiple-choice question format.
Database	Enables administrators to create, maintain and use data record entries.
Feedback	Allows educators to create and conduct online surveys, in order to gather student feedback.
Forum	Allows students and educators to have discussions.
Glossary	Educators can create a list of words relating to a specific subject or topic.
Lesson	Allows educators to create lessons and to present them in flexible ways.
Quiz	Provides a means for an educator to create online quizzes, which may be automatically marked.
Survey	Allows educators to create and conduct online surveys.

Wiki	A collection of web pages that allows anyone to modify the content collaboratively.
Workshop	Enables peer assessment.

Moodle provides educators with a means to generate action logs (Estacio & Raga, 2017). This allows educators to track the engagement of students with course resources as well as their participation in course activities. Data collected in the log files of Moodle include explicit student actions, such as viewing courses, submitting assignments and participating in quizzes, as well as posting on discussion forms, amongst others. Each event record in a raw action log has eight attributes, namely time, user ID, event context, component, event name, description, origin and IP address. Table 2.2 defines each of these attributes.

Table 2.2: Action Log Data Attributes

Data Attribute	**Description**
Time	When the action was executed (includes date and time).
User ID	User who initiated the action.
Event context	Item on which action was initiated.
Component	Type of activity initiated (e.g. quiz, assignment, etc.).
Event name	Type of action executed (e.g. quiz attempt viewed).
Description	Description of the initiated learning activities.
Origin	Where the action took place.
IP address	IP address of the device being used.

Monarch Media Incorporated (2010) explored the advantages and disadvantages of open source LMS, in particular of Moodle. They claim the following five advantages:

1. Site management and administrator tools;

2. Variety of user management options;
3. Registration and enrolment tools;
4. Course management and communication options; and
5. Ease of use.

However, they also state three problems and concerns with its implementation:

1. Lack of extensive customisation options;
2. Dependence on third-party plug-ins to create functionality; and
3. Lack of a fully featured skills development and management toolset.

By 2019 the first two problems stated no longer exist and have been addressed through updates. Although the advantages of Moodle outweigh the disadvantages, the third disadvantage highlights the fact that Moodle still lacks a tool that can effectively make sense of the data collected in its log files. This problem is still prevalent in 2019. Processes such as learning analytics and educational data mining aim to solve this problem by making sense of this collected data.

2.3 Learning Analytics

New analytical methodologies, particularly LA, have made making sense of educational data possible. LA tools can potentially utilise log data to provide crucial information on how learning processes occur throughout a student's interaction with the LMS (Yu & Jo, 2014). The majority of LMS are generic and do not provide the depth of extraction and aggregation that is required for various contexts (Ferguson, 2012).

The field of LA has emerged owing to the increased interest in how educational data can be used to improve the teaching and learning environment. LA is formally defined by Siemens and Long (2011, p. 34) as:

> *"*The measurement, collection, analysis, and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs*."*

Educators and students are enabled to make data-driven decisions regarding student success and retention through the prediction capabilities that are offered by various Learning Analytic tools (Olmos & Corrin, 2012; Smith, Lange, & Huston, 2012). May (2011) suggests that LA can be both descriptive and predictive. Descriptive means that the raw data obtained from educational databases can be converted into information, such as how often a student logs into their institution's LMS. Predictive, on the other hand, means that data, such as how often a student logs into an LMS, could be used together with other data to predict whether a student will pass or not. Both the descriptive and predictive information generated may help educators to identify at-risk students and to provide interventions (Dietz-Uhler & Hurn, 2013).

In order to understand LA and how it can be applied, Greller and Drachsler (2012) proposed a model, depicted in Figure 2.2 that considers six critical dimensions of LA.

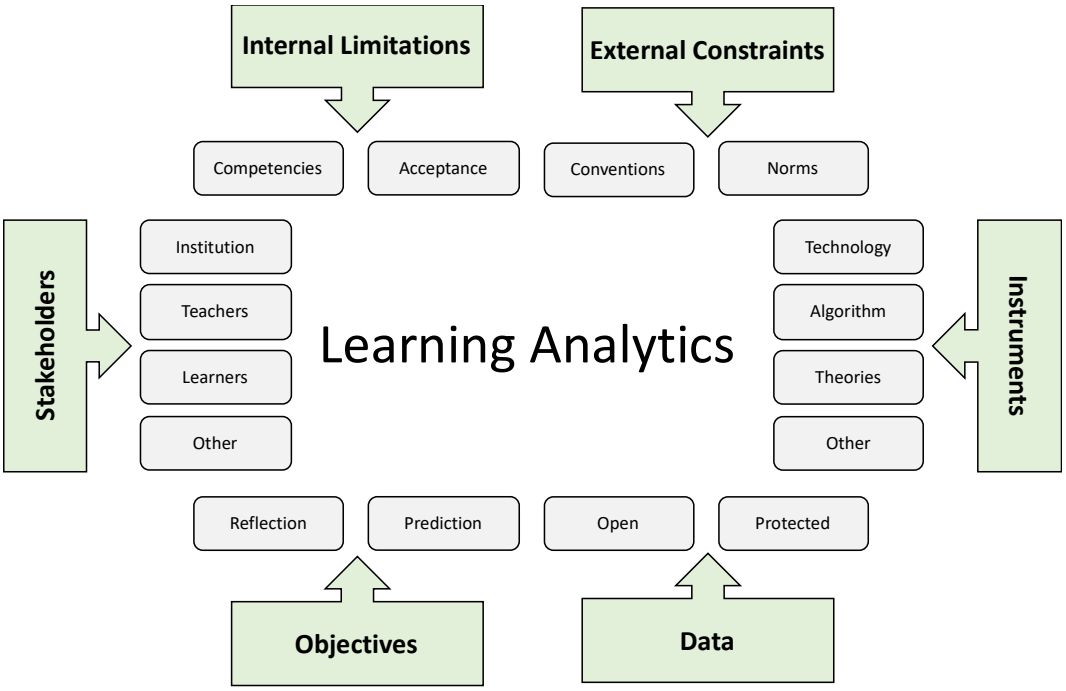

Figure 2.2: Critical Dimensions of Learning Analytics (Greller and Drachsler, 2012)

The first dimension, stakeholders, refers to two groups of people who are involved in the LA process: (1) the *data subjects*, which are the people on whom the data is based, such as the students (learners); and (2) the *data clients,* which are the people who will make use of this data such as educators (teachers).

The second dimension, objectives, explains what the intended use of the data is. As mentioned above, LA can be both descriptive and predictive, which relates directly to this dimension, as the objectives include reflection and prediction. Reflection on the data involves analysing student interactions and using that information accordingly. Prediction involves using the data to predict outcomes, which are not currently known.

The third dimension, data, describes what data is available to be used in order to achieve the above-mentioned objectives. This data can be openly available, such as

students' marks for assignments and tests, or protected, such as student interactions with an LMS.

The fourth dimension, instruments, is made up of various aspects included in the development of an LA system. Firstly, technologies that can be applied in the development of the system, such as machine learning. Secondly, algorithms, for example, the different types of machine learning techniques that can be used in the process. Some of these algorithms include neural networks, naïve Bayes classifiers and decision trees. Machine learning and each of these different techniques are discussed in detail in Chapter 3.

The fifth dimension, external constraints, refers to problems that may occur in the development of the system. Conventions can be separated into multiple aspects. One is privacy, which questions whether the analysis falls within the privacy arrangements of the institution and the law. Another is ethics, which questions the dangers of the abuse and misguided use of the data. Furthermore, norms question whether there are legal issues related to the use of student data. These external constraints are discussed in more detail in Section 2.6.

The final dimension, internal limitations, mentions that the data clients may not have the required competencies to interpret and act upon the results of the system, together with the fact that the results might not be accepted.

Another technique used to make sense of educational data is educational data mining (EDM). EDM and LA are closely related as they are similar in both process and results (Elias, 2011).

2.4 Educational Data Mining

Data mining is defined as a computational method of processing existing data with the aim of obtaining useful knowledge from that data (Klösgen and Zytkow, 2002). By 2012 there had been an increased interest in using data mining for educational purposes (Osmanbegović & Suljić, 2012). When data mining is applied in education, it is referred to as EDM. This form of data mining is one of the most popular techniques for predicting student academic performance (Hamsa, et al., 2016; Shahiri, et al., 2015). The amount of data stored in educational databases is increasing rapidly. These databases contain hidden information for the improvement of students' performance. Higher education institutes could, through using EDM, be enabled to make more effective decisions. These decisions may lead to an improved quality of learning instruction and services (Al-Twijri & Noaman, 2015).

The EDM process converts raw data gathered from educational systems into useful information. This information could potentially have a great impact on educational research and practice (Kaur, Singh, & Josan, 2015). EDM uses machine learning techniques such as decision trees, neural networks, naïve Bayes classifiers, and k-nearest neighbour, amongst others. Using these techniques, different types of knowledge can be discovered such as association rules, classifications and clustering (Baradwaj & Pal, 2012). Predicting a student's academic performances from an LMS is an ongoing educational challenge. However, EDM makes this

possible by enabling the prediction of an unknown variable, in this case a pass mark, that describes students (Osmanbegović & Suljić, 2012).

Similar to LA, there are various stakeholders or participants in the educational process that could benefit by applying data mining to education. In Figure 2.3, Romero and Ventura (2007) present the cycle of applying data mining to education, as well as the participants that are affected.

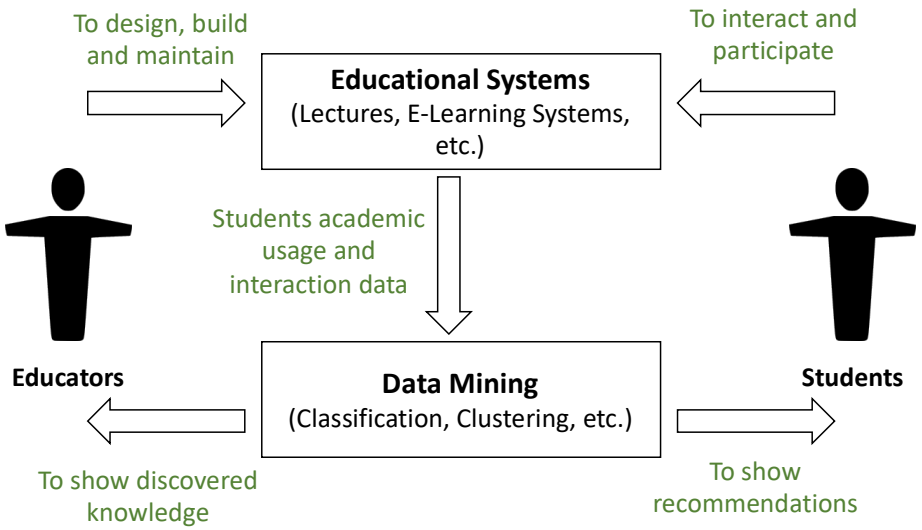

Figure 2.3: The Cycle of Applying Data Mining to Education (Romero and Ventura, 2007)

The cycle presented in Figure 2.3 begins with educators, who create some form of educational system with which students interact and participate. By doing so, large amounts of academic, usage and interaction data is created. This data is used in the data mining process in order to obtain knowledge previously unknown. This knowledge can be used by educators to maintain their educational systems, as well as to provide relevant interventions to the students. However, as mentioned in

Section 2.3, there are some external constraints and issues associated with LA and the EDM process.

2.5 Issues with Learning Analytics and Educational Data Mining

As mentioned in Section 2.3, and seen in Figure 2.2, two of the critical dimensions of LA and therefore EDM, are internal limitations and external constraints. Campbell, DeBlois and Oblinger (2007) provide a list of the issues and concerns that must be addressed before implementing any LA or EDM systems Some of these concerns are presented in Table 2.3.

Table 2.3: Issues and concerns regarding LA and EDM

Concern	Description
Big Brother	Students and educators may feel threatened to know that someone can track all that they do.
Holistic view	There is a concern that any data set, no matter how comprehensive, cannot take into account some issues, such as interpersonal ones.
Faculty involvement	Educators need to be involved in order for learning analytics to have its greatest impact.
Obligation to act	Are educators and institutions obligated to use data to increase the probability of student success?

Data privacy and the use of data are also strong concerns during LA and EDM processes. There are legal and ethical issues, such as the Family Educational Rights and Privacy Act (FERPA), that need to be addressed before faculty or institutions can make use of some student data (Campbell et al., 2007; Greller & Drachsler, 2012). FERPA is a law that protects the privacy of student educational records in

the USA. In South Africa, a similar law is present called the Protection of Personal Information Act (POPIA).

With EDM and LA clearly defined, it is evident that a fundamental aspect of both processes is the educational data required as input in order to make data-driven decisions.

2.6 Educational Data

From the above literature, it is evident that there are two main factors in predicting student academic performance, namely attributes and prediction techniques (Shahiri et al., 2015). This section discusses the attributes used to predict student performance. Data-driven decisions involve making use of data, such as that provided by an LMS, to create new knowledge and to inform educators (Long & Siemens, 2011). This data can be obtained from various sources.

The first, most obvious source of data generated by students in performing academic activities, is tests, assignments and practical assessment results. These summative assessment results are typically stored for each module. The next source of data is as a result of an increased popularity in e-learning. E-learning uses electronic technologies in order to distribute educational material to students (Fry, 2001). With the increased adoption of e-learning as a method for educators to share educational resources and material online, institutions are starting to use LMS as a preferred method of distribution (Unwin, et al., 2010). A key aspect of an LMS is

that it records all the interactions between a student and the system, with varying levels of granularity depending on the LMS (Liu, et al., 2015).

A rich dataset, along with numerous attributes, is the basis for advanced learning analytics and educational data mining (Daud et al., 2017). Educational data includes attributes such as submission time, daily interactions and time spent on assessments (Hamsa et al., 2016). These attributes all form part of the educational data that is relevant for prediction. By adding more relevant attributes, the accuracy can be increased (Hamsa et al., 2016). With this information, educators can provide interventions to students that will help them succeed (Long & Siemens, 2011). For example, if a student has not logged into their LMS for a certain period of time, this may suggest to an instructor that the student requires an intervention to improve their academic performance. Measuring the academic performance of students is challenging since the academic performance of students hinges on diverse factors like personal, socio-economic, psychological and other environmental variables (Ramesh, Parkavi, & Ramar, 2013). The following are some of the most common success factors used to predict student academic performance in related literature.

Performance Prediction Attributes

Shahiri et al. (2015) conducted a study that identified internal assessment results as being the most popular factor in predicting the academic performance of students. The internal assessments include attributes such as assignment marks, quiz marks, lab work, class tests and class attendance. Student demographics and external assessment attributes closely followed in popularity. The student demographics

include attributes such as student age and gender. External assessments are identified as attributes, such as the final examination mark for a particular subject related to the one currently being investigated. Psychometric factors, such as students' interests, study behaviours and family support have also been used to predict student academic performance but are not as useful owing to the qualitative nature of the data. However, some psychometric factors, such as student interactions with an LMS, can be useful as they are able to be stored as quantitative data.

Osmanbegović and Suljić's (2012) research findings contrast with the findings of Shahiri et al. (2015), as their research found that the gender of the student had the smallest impact on a student's academic performance. However, Kotsiantis, Pierrakeas and Pintelas (2004), and Shalem, Bachrach, Guiver and Bishop (2014) also concluded that demographic attributes have a high impact on predicting student academic performance. Family income and expenditure were also found to have a high impact on the prediction of the students' performance (Ramaswami & Bhaskaran, 2010; Osmanbegović & Suljić, 2012; Abu Tair & El-Halees, 2012).

Ramaswami and Bhaskaran (2010) conducted a study that made use of the CHAID prediction model to analyse the relationship between variables that can be used to predict the academic performance of students. Various attributes were identified as strong indicators. These include the student's marks obtained in secondary education, the living area of the student, the location of the student's school, and the type of secondary education enjoyed by the student.

The research by Smith, et al. (2012), on predicting student academic performance revealed that the frequency with which the student logs in to the LMS, how often the student engages with the course material, the learning pace of the student, and the student assignment marks successfully predicted their performance in the course.

Hijazi and Naqvi (2006) conducted a study on the performance of students. The study involved using simple linear regression analysis to predict student academic performance. Factors such as mother's education and student's family income were highly correlated with the academic performance of the student.

Baradwaj and Pal (2011) examined the performance of students in a course with the goal of predicting their performance in an examination that takes place at the end of the semester. The study concluded that attributes such as previous semester marks, class test marks, seminar performance, assignment marks, general proficiency, class attendance and laboratory work, all play an important role in predicting the performance of students.

2.7 Conclusion

This chapter introduced learning management systems with a focus on Moodle, together with providing a more in-depth understanding of learning analytics and educational data mining. The process and similarities for each of these techniques were discussed, together with some of the problems and issues commonly faced.

Finally, A literature review was performed in order to identify popular attributes used to predict student academic performance in related studies.

This chapter identified common attributes used to predict student academic performance and therefore addressed SO_1: *To identify common attributes of student activities used to predict academic performance.* Secondly, this chapter identified the data and attributes currently available in the Moodle LMS, therefore addressing SO_2: *To identify what data and attributes are currently available in Moodle to be used for predicting student group academic performance.* The attributes identified in Section 2.6, together with the available data within Moodle discussed in Section 2.2, are used to produce the dataset. This dataset is used by the relevant machine learning techniques used in learning analytics and educational data mining as they were defined in this chapter.

Chapter 3 introduces the concepts of data pre-processing and machine learning. The process of data pre-processing is described in detail. A literature review is performed to identify popular machine learning techniques used to predict student academic performance in related studies. Each of the selected techniques is then discussed in detail.

Chapter 3: Machine Learning

3.1 Introduction

Chapter 2 focused on the literature surrounding learning management systems (LMS), educational data mining (EDM) and learning analytics (LA). Various performance prediction attributes were identified for possible use in the prediction models. This chapter continues developing the objectives of the study by providing insight into the processes that occur before, during and after the modelling process. The findings of this chapter will assist in planning for the requirements and subsequently, for the design of the final predictive models. The layout and research objectives of the chapter are illustrated in Figure 3.1. The following research objective (Section 1.3) is addressed in this chapter:

SO_3: *To determine the machine learning techniques commonly used by educational data mining and learning analytics and argue their relevance to a specific data set.*

Before modelling can be done, the data must first be pre-processed (Section 3.2). Machine learning is discussed in order to gain an understanding of the process involved (Section 3.3) Additionally, literature is reviewed to identify commonly used machine learning techniques (Section 3.4) and these are discussed in detail (Section 3.5 – 3.7). Finally, different methods of evaluating machine learning techniques are discussed (Section 3.8).

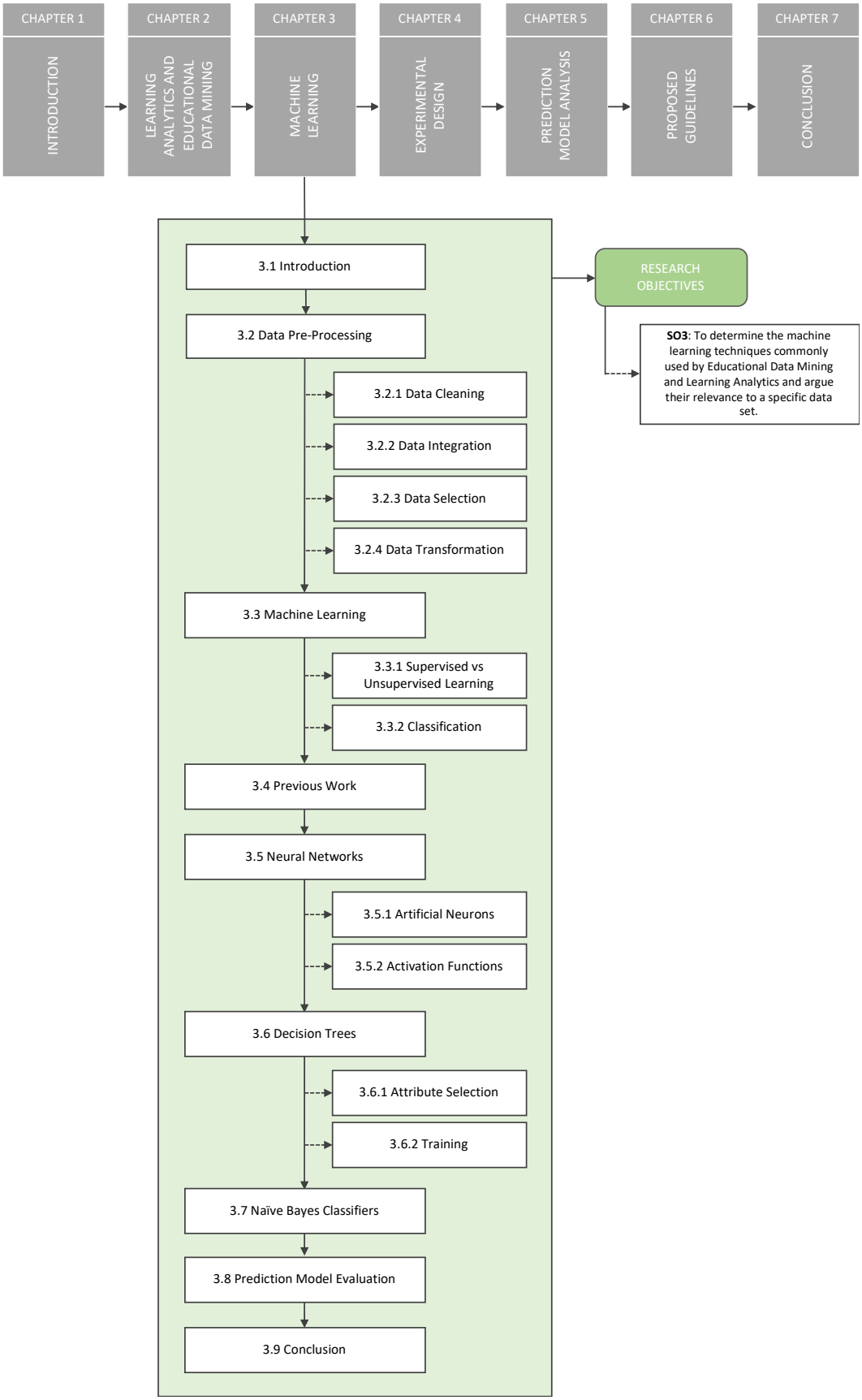

***Figure 3.1**: Chapter Three Outline*

3.2 Data Pre-Processing

Han, Kamber and Pei (2012) define data pre-processing as the set of techniques used prior to the application of a data mining method. Figure 3.2 depicts the entire prediction modelling process. Data pre-processing is one crucial step in the modelling process (Shahiri, et al., 2015). The types of phases performed during this step may vary from dataset to dataset, but can be grouped into four general categories (Han, et al., 2012), namely data cleaning (Section 3.2.1), data integration (Section 3.2.2), data selection (Section 3.2.3) and data transformation (Section 3.2.4).

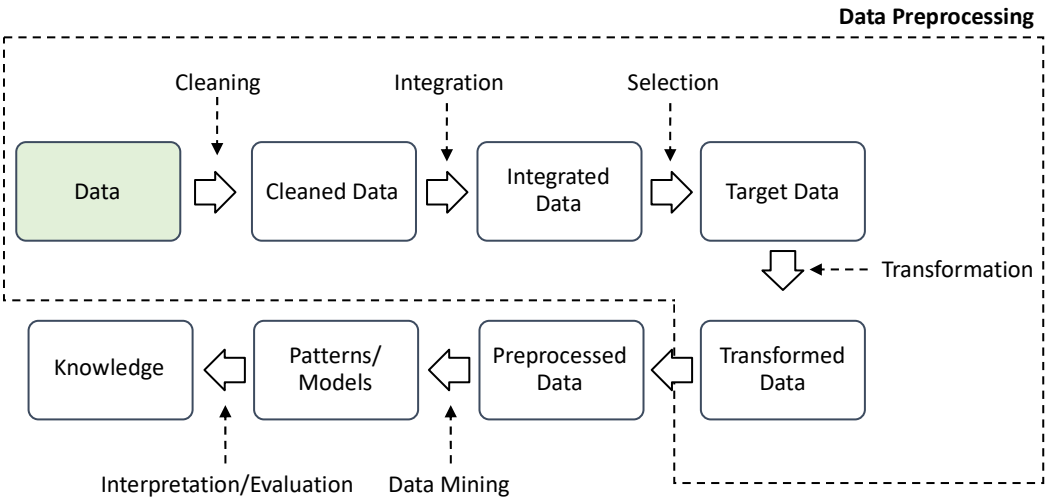

Figure 3.2: Prediction Modelling Process

Data pre-processing is an essential part of any prediction modelling process owing to its capability to convert data to a format that the data mining algorithm is able to process (Uma & Hanumanthappa 2017). In the event that the accumulated data is not appropriate to allow information to be generated from it, prediction will be more

challenging, if not impossible (Kotsiantis and Kanellopoulos, 2006). Lee, Hsu, and Kothari (2004) reinforce this notion by referring to the 'garbage in, garbage out' principle and how clean data is crucial for data mining.

Before any data pre-processing can take place, relevant data needs to be collected. For the context of this study, educational data in the form of Moodle log files and assessment mark files are appropriate, as described in Section 2.6. Once the data has been collected, data pre-processing may begin with the first phase being data cleaning.

3.2.1 Data Cleaning

Most techniques in data mining rely on a data set that is complete and noise free. Data cleaning deals with detecting and removing errors and inconsistencies from data in order to improve the quality of that data (Rahm & Do, 2000). Figure 3.3 represents this first phase of data pre-processing.

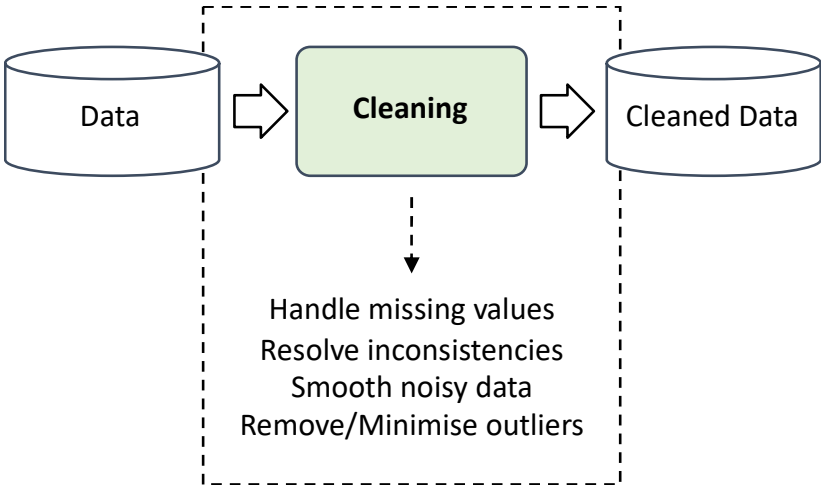

Figure 3.3: Data Cleaning Process

During data cleaning, various techniques are employed to clean the data by filling in missing values, smoothing noisy data, identifying or removing outliers and resolving inconsistencies (Uma & Hanumanthappa 2017; Han, et al., 2012).

Missing values are values that are not present in the collected data. Missing data can be handled in several ways, such as removing the observation, filling the value with the mean value of the column, or replacing the value with a global constant (Han, et al., 2012).

Noisy data occurs when inconsistencies occur in the data recording process which, in turn, affects the interpretability of the data (Van Hulse & Khoshgoftaar, 2009). It is essential that data is analysed to determine what kind of noise, if any, appears in the data set, and to apply appropriate methods to remove or minimise this noise.

Outliers are values that fall outside of a set of clusters (Han, et al., 2012). Outliers come in three general forms, namely extreme random variability in the data, gross deviation from experimental procedure and data capturing or calculation error. The first type of outlier mentioned should be retained and processed along with all the other data points. The second and third types of outlier need to undergo further investigation. This investigation would ultimately determine whether the data point should be kept or discarded (Grubbs, 1969).

Once data cleaning has been performed, the next phase of data pre-processing is to integrate the data.

3.2.2 Data Integration

Data integration involves integrating and merging multiple data sources into one. Careful integration can help to reduce redundancies and inconsistencies in the resulting data set. Figure 3.4 represents the data integration phase of data pre-processing.

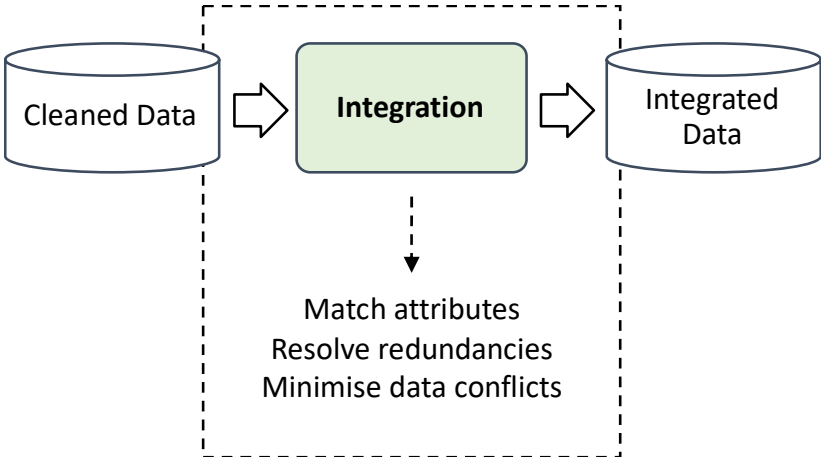

Figure 3.4: Data Integration Process

There are several issues to consider during data integration. It is essential that careful entity identification is done in order to ensure that attributes from different data sources are correctly matched. One such case could be Student name in one data source consisting of only the name and surname of the student; and Name in another source consisting of only the first name of the student. Special attention should also be paid to the structure of the data in order to ensure that any attribute functional dependencies and referential constraints in the source data set match those in the target data set.

Data redundancy is another issue that may arise during data integration. Any attribute that can be derived from another attribute or set of attributes is redundant. The integrated data set should be free of any data redundancies.

Owing to variations in representation, scaling or encoding, attribute values from different sources may differ (Han, et al., 2012). Data integration also involves the detection and resolution of data value conflicts. An attribute being stored as a percentage in one data source and as a total in another source, is one such example.

Once the data from multiple data stores has been integrated, the next step in data pre-processing is to select the data to be used by the selected machine learning techniques.

3.2.3 Data Selection

Large quantities of historical data are collected in data repositories, but usually not all of it is required. Therefore, data selection needs to take place in order to retrieve the relevant data from the database. Applying data selection techniques results in obtaining a reduced representation of the data set that closely maintains the integrity of the original data. The produced data set should be more efficient and should produce the same analytical results as the original data set. Figure 3.5 represents the data selection phase of data pre-processing.

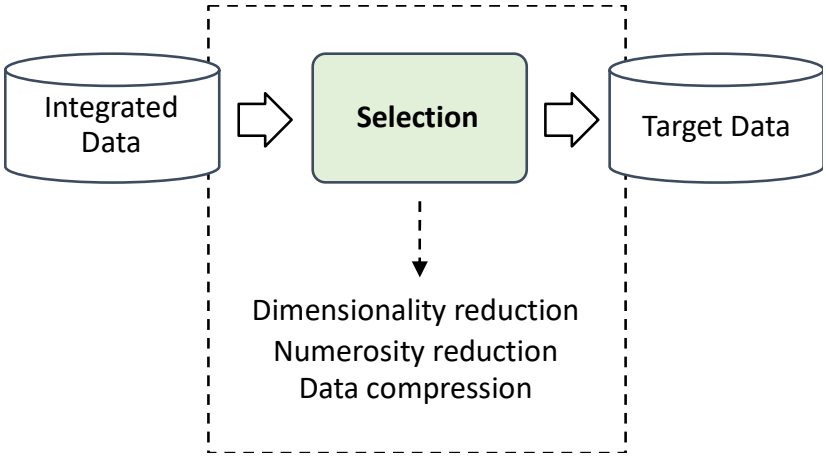

Figure 3.5: Data Selection Process

Data selection strategies include dimensionality reduction, numerosity reduction and data compression (Han, et al., 2012). These strategies are described in Table 3.1.

Table 3.1: Data Selection Strategies

Strategy	Description
Dimension reduction	The process of reducing the number of variables by eliminating irrelevant and redundant attributes.
Numerosity reduction	The use of parametric or non-parametric techniques to replace the original data volume by a smaller form of data representation.
Data compression	Lossless or lossy reduction is performed to compress the data. Lossless reduction allows the original data to be reconstructed from the compressed data without any loss, whereas lossy reduction can only approximate original data.

Once the data has been selected, the final step in pre-processing is to transform the data into the format required as input for the selected machine learning techniques.

3.2.4 Data Transformation

Data transformation is the final process of converting data into the format required by the data mining algorithm (Uma & Hanumanthappa, 2017). The transformed data results in a more efficient mining process and patterns that are easier to understand. Figure 3.6 represents the data transformation phase of data pre-processing.

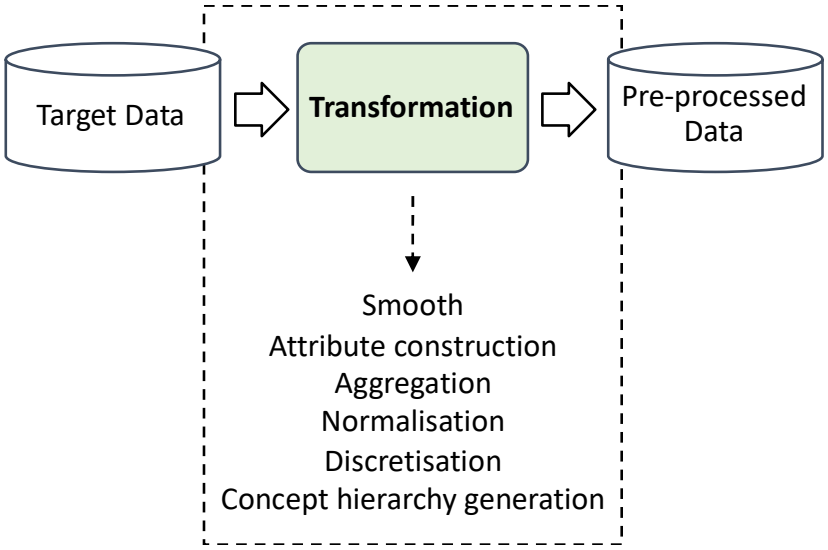

Figure 3.6: Data Transformation Process

Strategies for data transformation include those as described in Table 3.2 (Han, et al., 2012.

Table 3.2: Data Transformation Strategies

Strategy	Description
Smoothing	Binning, regression and clustering techniques are used to remove noise from data.
Attribute construction	New attributes are constructed from the given set of attributes.
Aggregation	Aggregated operations are applied to the data.
Normalisation	Attribute data is scaled to fall within a smaller range.
Discretisation	Numeric attributes are replaced by interval labels or conceptual labels.
Concept hierarchy generation for nominal data	Attributes are generalised to higher-level concepts.

Once the data is completely pre-processed, it is ready to be used as input for the training and testing of selected machine learning techniques.

3.3 Machine Learning

The mining process described by Romero, Ventura and Garcia. (2008), as shown in Figure 3.2, is a two-phase process, which includes the initial pre-processing of data (Section 3.2) followed by the application of data mining algorithms that transform the data into a form suitable for interpretation and evaluation. These data mining algorithms are some form of machine learning.

According to Mitchell (1997, p. 2), machine learning can be formally defined as:

> "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."

Machine learning investigates how computer programs automatically learn to recognise complex patterns and make intelligent decisions based on data (Han, et al., 2012). Essentially, machine learning is the ability of a computer to learn from experience (Mitchell, 1997). The computer 'learns' by analysing large amounts of data with the goal of finding hidden patterns and rules. Recognising these patterns would be extremely difficult, if not impossible, for a human to do owing to the sheer volume of data. However, these patterns are mathematical in nature and can therefore easily be recognised and processed by a computer. Once the computer has developed the rules, it is able to characterise new, unseen data, meaningfully.

Machine learning is used in a wide range of areas. For example, search engines use machine learning to better relate search entries and web pages. By analysing the content of the web pages, search engines can identify which phrases are most often occurring in a web page and to use that information to return the most relevant results for a search phrase (Witten, Frank, Hall & Pal, 2016). Image recognition also uses machine learning to identify certain objects of an image, such as faces (Alpaydin, 2004). First, the machine learning algorithm analyses images that contain a certain object. The algorithm is then able to determine whether an image contains that object or not (Witten et al., 2016). In addition, machine learning can be used to predict the kind of products that a customer might be interested in. By

analysing the past products that a user has bought or viewed, the machine learning algorithm can make suggestions about other products that the customer might be interested in (Witten et al., 2016). There are two types of machine learning techniques, namely supervised and unsupervised.

3.3.1 Supervised versus Unsupervised Learning

Depending on the type of input data, machine learning algorithms can be divided into supervised and unsupervised learning.

Supervised learning algorithms learn on a labelled data set. This allows the algorithm to evaluate its accuracy on training data (Mohri, Rostamizadeh & Talwalkar, 2012; Mitchell, 1997). These algorithms usually perform classification activities that can predict one property by using other properties. It is imperative that the class structure of the input data is the same as that of the data required to be processed.

Unsupervised learning algorithms, in contrast, make use of unlabelled data that the algorithm attempts to understand by extracting patterns and features of its own (Sugiyama, 2015; Mitchell, 1997). These algorithms usually perform clustering activities (Han, et al., 2012).

This research focuses on supervised learning, specifically classification, as the data set contains a target class (desired output) associated with each input vector.

3.3.2 Classification

Classification is a supervised learning method, used to analyse the attributes of data items and to assign each of the data items to a class or category (Ahmed & Elaraby, 2014). Han, et al. (2012), list some of the numerous applications of classification, namely:

- Fraud detection
- Target marketing
- Performance prediction
- Medical diagnosis

As mentioned above, Han, et al., (2012) name performance prediction as one of the applications of classification models. Predictive analytics makes use of techniques such as data mining, machine learning and statistics to build a model from past data, to make predictions about future events and behaviours present in previously unseen data (Nyce, 2007; Shmueli & Koppius, 2011). Predictive analytics is evident in fields such as, education, healthcare, finance and law (SAS, 2017).

A prediction model typically makes use of two separate sets of data. The first set, which is used to train the prediction model, is called the training set. This data set consists of known data, and is fed into the model to allow it to identify patterns and rules. The testing set is then used to determine the accuracy of the model and to ensure that it is flexible enough to be used on unseen data sets. Adjustments can then be made on the model based on the results of the testing. It is vital that two

different sets are used in order to identify issues such as overfitting or underfitting. Overfitting may occur when a model is accurate with its original data set but performs poorly on unseen data sets. Underfitting occurs when the learning algorithm cannot capture the underlying trend of the data (Srivastava, Hinton, Krizhevsky, Sutskever & Salakhutdinov, 2014). In order to identify relevant machine learning techniques that would perform best when used in this study, a literature review was performed on existing work.

3.4 Related Work

Applying classification techniques to predict student group academic performance is a challenging task. The initial problem that needs to be addressed is identifying the best techniques to be used (Shahiri et al., 2015).

Nghe, Janacek and Haddawy (2007) compared two machine learning techniques, namely decision trees and Bayesian networks. They used student records and grade point averages of the previous year to predict performance. Decision trees achieved an accuracy of 94.03% and Bayesian networks an accuracy of 90.27%.

Zekić-Sušac, Frajman-Jakšić and Drvenkar (2009) used decision trees and neural networks for predicting student performance. They used student gender, whether the student had a scholarship, time dedicated to studying, and class attendance to predict performance. Decision trees achieved an accuracy of 88.36% and neural networks an accuracy of 66.26%.

Shah (2012) compared the accuracy of different classifiers in predicting student academic performance, namely decision trees and Bayesian networks. Decision trees achieved an accuracy of 92% and Bayesian networks an accuracy of 59%.

Herzog (2006) used decision trees and neural networks, in order to predict student failures and degree-completion time. He compared the accuracy of decision tree algorithms and three artificial neural networks. The decision tree algorithm achieved the highest accuracy of 93% and the best neural network an accuracy of 85%.

Simeunović and Preradović (2014) used logistic regression, decision trees and neural networks to predict student academic performance. They used student demographic data, behavioural data, and grade point averages to predict student academic performance. Decision trees achieved an accuracy of 71.25%, logistic regression an accuracy of 74.80%, and neural networks an accuracy of 76.40%.

Ibrahim and Rusli (2007) used neural networks, decision trees and linear regression to predict student academic performance. They used demographic data and grade point averages to predict performance. This study used root mean square error (RMSE) to measure accuracy. Decision trees achieved an RMSE of 0.1769, neural networks 0.1741 and linear regression 0.1848.

Cheewaprakobkit (2015) used decision trees and neural networks in order to classify students based on their academic performance. Decision trees achieved an accuracy of 85.19% and neural networks an accuracy of 83.88%.

Osmanbegović and Suljić (2012) compared decision trees, neural networks and Bayesian classifiers in predicting student academic performance. They used student gender, high school results, grade point averages and earnings. The Bayesian networks achieved an accuracy of 76.65%, decision trees an accuracy of 73.93% and neural networks an accuracy of 71.20%.

Table 3.3 shows a summary of the related research discussed in this section.

Table 3.3: Summary of Previous Research

Author	Year	Methods	Results
Nghe et al.	2007	Decision trees, Bayesian networks	**Decision tree: 94.03%** Bayesian network: 90.27%
Zekić-Sušac et al.	2009	Decision trees, Neural networks	**Decision tree: 88.36%** Neural network: 66.26
Shah	2012	Decision trees, Bayesian networks	**Decision tree: 92%** Bayesian network: 59%
Herzog	2006	Decision trees, Neural networks	**Decision tree: 93%** Neural network: 85%
Simeunović & Preradović	2014	Decision trees, Neural networks, Logistic regression	Decision tree: 71.25% **Neural network: 76.40%** Logistic regression: 74.8%
Ibrahim & Rusli	2007	Decision trees, Neural networks, Linear regression	RMSE: Decision tree: 0.1769 Neural network: 0.1714 **Linear regression: 0.1848**
Cheewaprakobkit	2015	Decision trees, Neural networks	**Decision tree: 85.19%** Neural network: 83.88%

Osmanbegović & Suljić	2012	Decision trees, Neural networks, Bayesian networks	Decision tree: 73.93% Neural network: 71.20% **Bayesian network: 76.65%**

From this literature review it is evident that the most popular machine learning models used in order to predict student academic success are decision trees, naïve Bayes classifiers, neural networks, and logistic and linear regression. Of these, the three that achieved the highest accuracies were neural networks, decision trees and naïve Bayes classifiers. Therefore, this research focuses on the use of these three models.

3.5 Artificial Neural Networks

Artificial neural networks (ANN) are data models inspired by our understanding of the human brain (Freeman & Skapura, 1991). ANNs have three interconnected layers, an input layer (x_1, x_2,..., x_m), one or more hidden layers, and an output layer (y). Neural networks have established themselves as popular models for non-linear and classification problems (Tang & Salakhutdinov, 2013). Figure 3.7 provides a visual representation of a generic ANN.

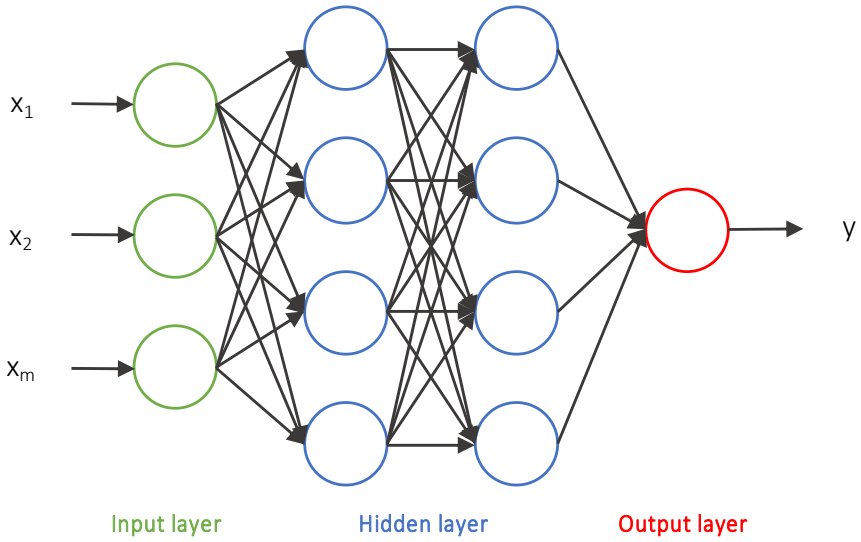

Figure 3.7: Artificial Neural Network Visualisation

ANNs have been used for a wide range of applications, including:

- Diagnosis of diseases
- Speech recognition
- Data mining
- Composing music
- Image processing
- Classification
- Pattern recognition
- Compression

A neural network, when used for classification, is typically a collection of neuron-like processing units with weighted connections between the units (Han et al., 2012). These processing units are called artificial neurons.

3.5.1 Artificial Neurons

An artificial neuron is a model of a biological neuron. Figure 3.8 provides a visual representation of an artificial neuron. Each neuron receives signals from the environment (x_1, x_2,..., x_m), or outputs from other neurons. Each of these signals is inhibited through some numerical weights (w_1, w_2,..., w_m), associated with each of the signals. These weighted signals are then summated and output. In summary, a neuron receives a set of inputs, calculates a weighted sum of the inputs, and then decides whether it should be activated or not, based on the result. The strength of the resulting signal (y) is controlled by a function known as the activation function.

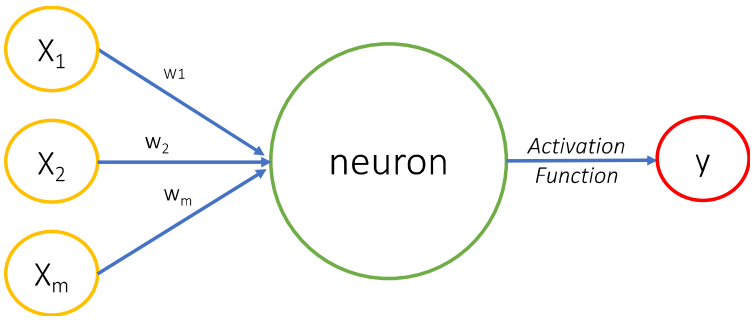

Figure 3.8: Artificial Neuron Visualisation

3.5.2 Activation Functions

An activation function defines the strength of the output signal of an artificial neuron. This section defines three activation functions, namely threshold function, sigmoid function and rectifier functions.

3.5.2.1 Threshold Function

The threshold activation function is a very simple activation function. This function is given here:

$$\emptyset(x) = \begin{Bmatrix} 0 \; if \; x < 0 \\ 1 \; if \; x \geq 0 \end{Bmatrix}$$

It receives the weighted summation from the neuron and if that value is less than 0, it outputs 0; if the value is greater than 0, it outputs 1. This function is visualised in Figure 3.9 where y is the output and x is the value of the weighted sums (Glorot, Bordes & Bengio, 2011).

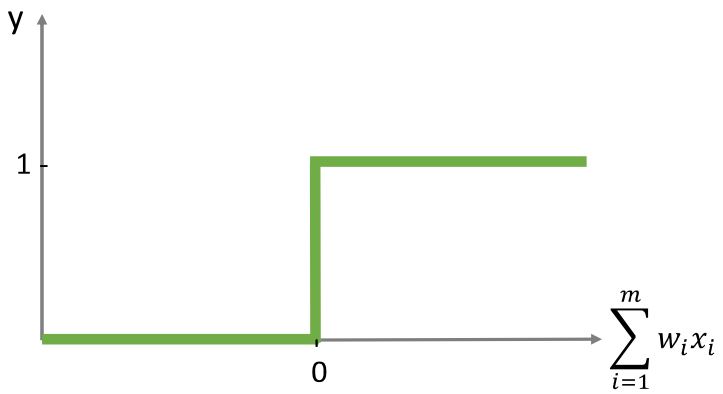

Figure 3.9: Threshold Activation Function (Glorot, et al., 2011)

3.5.2.2 Sigmoid Function

The sigmoid activation function is a more complex activation function. This function is defined as:

$$\emptyset(x) = \frac{1}{1 + e^{-x}}$$

The function allows for gradual progression of outputs as opposed to the threshold function, which is binary. This function is visualised in Figure 3.10, where y is the output and x is the value of the weighted sums.

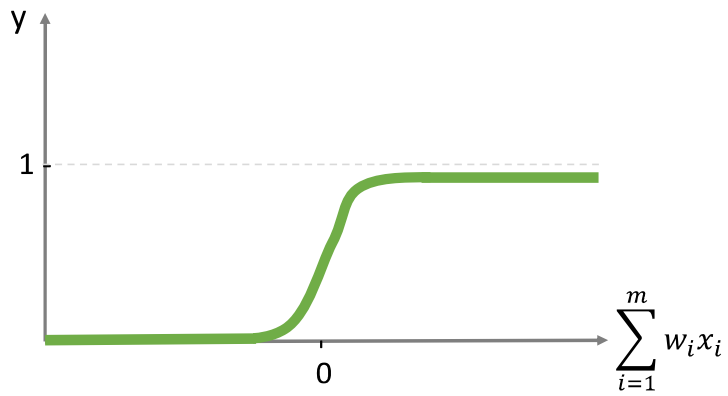

Figure 3.10: Sigmoid Activation Function (Glorot, et al., 2011)

The sigmoid function is useful in the final, output layer of the artificial neural network as it allows the prediction of probabilities (Glorot et al., 2011). For example, in this study, the probability that a student will fail is the required output. Therefore, this function is used by this study as the output function of the ANN prediction models.

3.5.2.3 Rectifier Function

The rectifier function is one of the most popular activation functions in ANN (Glorot, et al., 2011). This function is defined as:

$$\emptyset(x) = \max(x, 0)$$

Any value below or equal to 0, outputs 0, and for any value greater than 0, the output gradually progresses as the input value increases. This function is visualised in Figure 3.11, where y is the output and x is the value of the weighted sums.

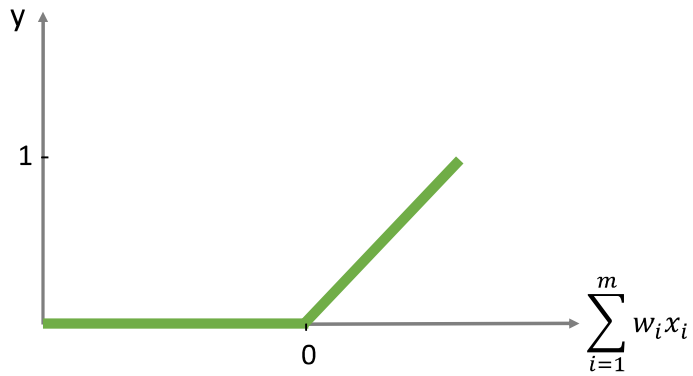

Figure 3.11: Rectifier Activation Function (Glorot, et al., 2011)

This function will be used by this study for the hidden layers of the ANN prediction models as it the most widely used activation function.

This section gave an overview on artificial neural networks and argued their relevance to this study. In the following section the focus will be on decision trees, the next machine learning technique to be discussed.

3.6 Decision Trees

A decision tree follows the same approach as humans do when making decisions (Do, 2007). The general idea behind a decision tree is to create a model, which can be used to predict a target class or value. It does so by implementing decision rules gathered from training data. The algorithm uses a tree representation to attempt to solve the problem.

The general structure of a decision tree consists of a root node. The root node has two or more subsequent child nodes. These child nodes can either be internal nodes or leaf nodes. A child node can then have its own children. Finally, a leaf node, also known as a decision node, has no child nodes attached to it and stores the final output for that path (Rokach & Maimon, 2005). Figure 3.12 is a simple example of a decision tree structure. Decision trees have several reasons that make them popular machine learning techniques (Khoonsari & Motie, 2012), namely:

- They are easy to understand.
- They are simple to implement.
- They can deal with both numerical and categorical data.
- They do not require data normalisation.

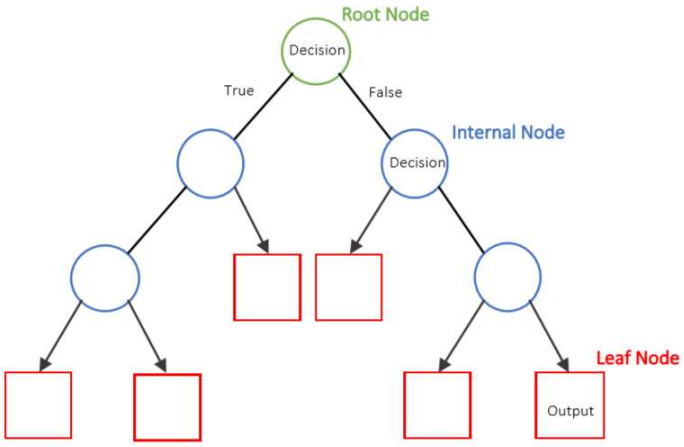

Figure 3.12: Decision Tree Visualisation

The main challenge during the creation of a decision tree is to identify the attributes that need to be considered at the root node and at each subsequent level. This is known as attribute selection.

3.6.1 Attribute Selection

An attribute selection measure is a measurement for selecting the best splitting criteria (Han, et al., 2012). Two popular attribute selection measures are information gain and Gini index.

Information gain chooses the attribute with the highest information gain as the splitting attribute. By using information gain as a measure, it estimates the information contained in each attribute. This measure chooses the attribute that has the maximum information gain (Quinlan, 1990). This approach minimises the tests needed to classify a given dataset (Han, et al., 2012).

The Gini index is similar to information gain; however, it will result only in binary splits (Patil, Lathi, & Chitre, 2012). The Gini index measures how often a randomly chosen element would be incorrectly identified. This results in the attribute with the lower Gini index being selected as the splitting attribute. These attribute selection algorithms form part of the training process.

3.6.2 Training

In order to learn, a decision tree follows these steps. All observations start at the root node of the tree. Using the Gini index or information gain measure, the attributes are considered to determine how the data can be split in the most informative way. This process continues until the split results in a single class.

The condition at the root node of the tree is always the most informative condition. When predicting an output value for a given dataset, the value of each attribute is

compared, using an 'if – then' test, with the corresponding attribute in the node, starting from the root. The branch corresponding to that value is followed to the next node. This process continues until a leaf node is reached with the predicted output value (Han, et al., 2012).

This section gave an overview on decision trees and argued their relevance to this study. In the following section the focus will be put on naïve Bayes classifiers, the final machine learning technique to be discussed.

3.7 Naïve Bayes Classifiers

Bayesian classifiers are statistical classifiers that can predict the probability of a given input belonging to a given target class. Studies found that the naïve Bayes classifier is comparable in performance to both decision tree and neural network classifiers (Kamber, et al., 2012). The naïve Bayes classifier is considered to be 'naïve' as it assumes that the effect of an attribute's value on a given class is independent of the values of other attributes (Bishop, 2006). Bayesian classifiers are based on Bayes' theorem.

Bayes' theorem is named after Thomas Bayes, who worked in probability theory. This theory states that a given data tuple X and some hypothesis H, such as X belongs to class C. We want to determine P(H|X), which is the probability that H is true given the observed data tuple X. P(H|X) is known as the posterior possibility of H given X.

In contrast, P(H) is known as the prior probability of H. This is the probability of H regardless of the value of X. Similarly, P(X) is the prior probability of X. This is the probability of X regardless of H. The posterior probability P(X|H) is the posterior probability of X given H.

Bayes' theorem provides a way of calculating the posterior probability, P(H|X) from the probabilities P(H), P(X) and P(X|H). Therefore, Bayes' theorem is defined by the algorithm:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

Given a dataset, the naïve Bayes classifier predicts the probability over a set of target classes rather than the exact class. There are three types of naïve Bayes classifiers, namely: Gaussian, multinomial and Bernoulli.

Gaussian is used when all the features are continuous. Multinomial is used when the features contain discrete data such as ranges. Bernoulli assumes that all the features of a given dataset are binary. A Gaussian naïve Bayes classifier is used in this study, owing to its ability to allow each feature to have different values of varying types.

One of the advantages of a naïve Bayes classifier is that it is not sensitive to irrelevant features in a given dataset. However, a disadvantage is that it assumes that each of the features is independent of the other, which is not always the case.

This section gave an overview on naïve Bayes classifiers and argued their relevance to this study. With each of the selected machine learning techniques discussed, the following section will describe the various performance metrics that could be used to evaluate the prediction models created from each of the machine learning techniques.

3.8 Prediction Model Evaluation

There are widely used indicators for evaluating the effectiveness of machine learning algorithms, such as precision, recall and f-measure (Powers, 2011).

In order to evaluate the effectiveness of a prediction model, predicted values must be compared with actual values.

Table 3.4: Confusion Matrix

	Predicted Fail (0)	Predicted Pass (1)
Actual Fail (0)	True Positive	False Negative
Actual Pass (1)	False Positive	True Negative

Table 3.4 is a matrix that shows the possible prediction results. It is called a confusion matrix. In this research there are two classification classes, fail and pass, represented by binary values 0 and 1 respectively. From the confusion matrix above, the following deductions can be made:

- True positive (TP): Correctly predicted that a student did fail.
- True negative (TN): Correctly predicted that a student did pass.
- False positive (FP): Incorrectly predicted that a student did fail.

- False negative (FN): Incorrectly predicted that a student did pass.

There are different evaluation criteria that can be obtained from these values. One is accuracy, defined by Powers (2011) as follows:

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

Accuracy is the ratio of correct predictions. However, accuracy has limitations in evaluating the prediction performance. One such limitation is when the class distribution is imbalanced. Accuracy does not show how the cases of minority class are classified, for example, a dataset that contains 100 students, 90 of whom passed. A crude prediction can be made and can achieve an accuracy of 90%. A crude prediction does not use any machine learning methods, but instead predicts that every student will pass. The model should perform better than just guessing that each case belongs to the majority class.

In this study, three other criteria are used. Precision and recall are defined by Powers (2011) as follows:

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

$$Recall = \frac{TP}{TP + FN}$$

The main idea is that predicting a positive outcome accurately is not enough. A good predictive model must have a good combination of successful positive predictions and successful negative predictions.

The third criteria that is used by this study is called F-measure, and it is defined by Powers (2011) as follows:

$$F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

F-measure is a way of having a single value that takes both precision and recall into account. F-measure is the final evaluation criterion for comparisons on the selected machine learning models in this research.

3.9 Conclusion

This chapter introduced the concepts of data pre-processing and machine learning. The process of data pre-processing was described in detail together with each of the steps involved. A literature review was performed in order to identify popular machine learning techniques used to predict student academic performance in related studies. Artificial neural networks, decision trees and naïve Bayes classifiers were selected. Each of these selected techniques was then discussed in detail. Finally, various evaluation criteria were discussed and the final criterion to be used by this study was identified.

The machine learning techniques identified in this chapter were used in Chapter 4 to create prediction models. Therefore, SO_3 was addressed: *To determine the machine learning techniques commonly used by educational data mining and learning analytics and argue their relevance to a specific data set.* These techniques were neural networks, decision trees and naïve Bayes classifiers. Before this could be done, the attributes identified in Chapter 2 needed to be pre-processed using the

process described in Section 3.2. These prediction models were evaluated in Chapter 5 using the evaluation criteria discussed in Section 3.8. These evaluation criteria will assist in addressing SO_5: *To evaluate and compare the performance of selected machine learning techniques in predicting student group academic performance.*

Chapter 4 uses the literature gathered in Chapter 2 and Chapter 3 to design and perform experimentation. The data gathered is pre-processed in preparation to be used as a dataset by the machine learning techniques selected in this chapter in order to create the various prediction models requiring analysis and comparison.

Chapter 4: Experimental Design

4.1 Introduction

The previous chapters discussed the literature necessary to create the prediction models. Having introduced learning analytics (LA) (Section 2.3) and educational data mining (EDM) (Section 2.4), the data pre-processing process was described (Section 3.2) and the relevant machine learning techniques were selected and discussed (Section 3.4 - 3.7), in order for the experimentation to begin. The first step of exploratory experimentation is experiment design. The design of this research is the key focus of this chapter. The use of the selected machine learning techniques to create prediction models is key to this research study and is the focus of this chapter. The layout and research objectives of the chapter are illustrated in Figure 4.1. This chapter discusses and addresses the following research objective:

> SO_4**:** *To use the selected machine learning techniques to create prediction models to predict student group academic performance.*

The chapter begins by reiterating and describing the purpose of the study (Section 4.2). Once a thorough understanding of the purpose is developed, data pre-processing techniques are applied and discussed in terms of each raw data set (Section 4.3). The prediction models are then developed using the pre-processed data set (Section 4.4).

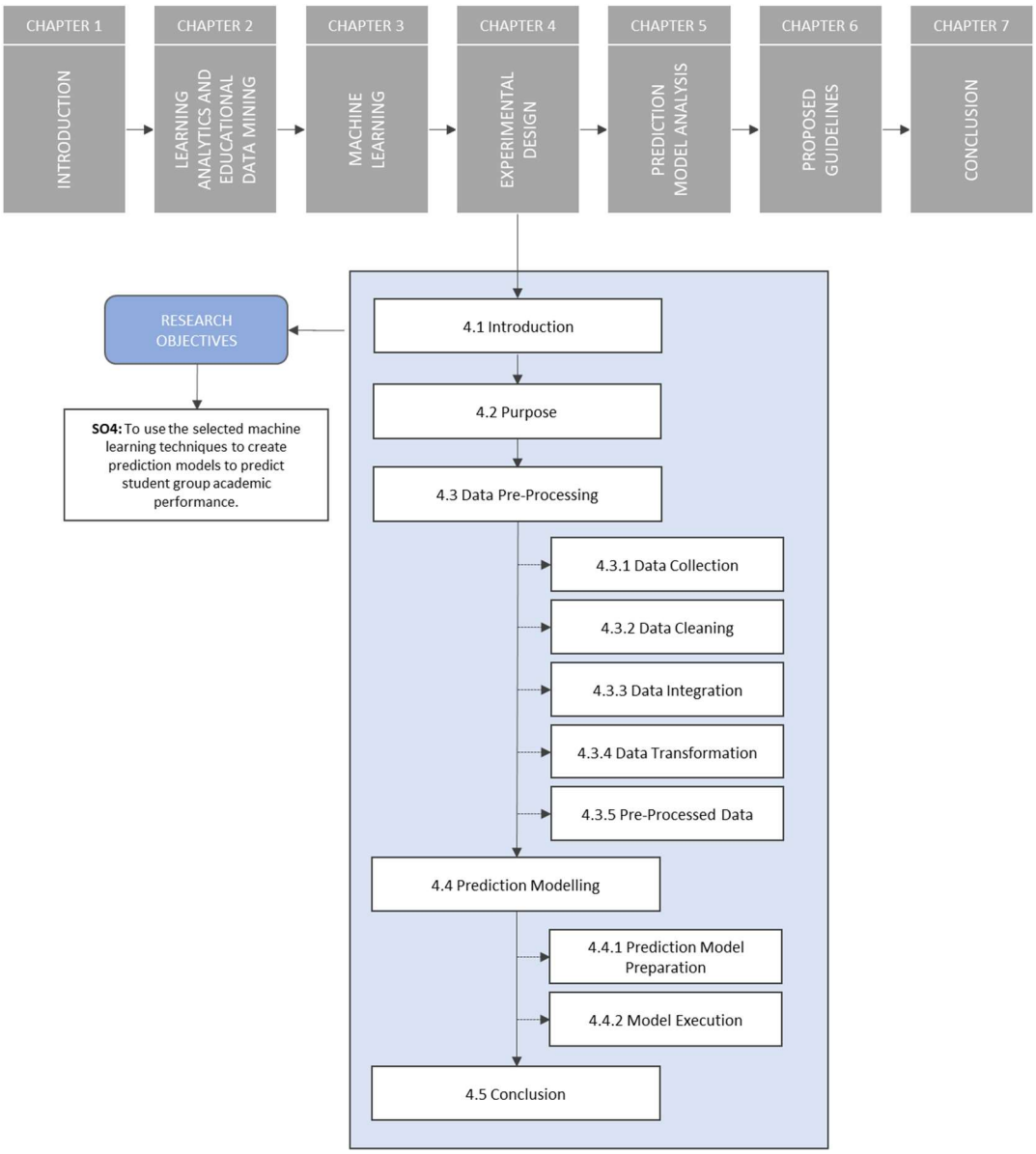

Figure 4.1: Chapter Four Outline

4.2 Purpose, Setting and Context

As stated in Section 1.3, the purpose of this research is to develop guidelines for the use of machine learning to predict student group academic performance. In order to develop these guidelines, three different prediction models were developed, and various experiments were conducted on each of the models (Section 4.4). Each

prediction model was trained and tested on three different data sets, one for each of the first three quarters of the year; Quarter 1, Quarter 2 and Quarter 3. These three datasets correspond with the first three of the four iterations of the third-year capstone project at the Nelson Mandela University. Data from the fourth quarter was not used as this stage of the year was deemed to be too late for effective intervention. The results of these experiments are provided and discussed for each selected machine learning technique (Section 5.3), and against each of the others (Section 5.4). Various performance metrics (Section 3.7) are used to compare the performance of the different prediction models.

The third-year capstone project consists of four iterations of assessment throughout the year, one per term. In addition to these assessments, students are required to submit weekly deliverables via Moodle. The data used for this study was therefore in the form of Moodle action logs generated by the students interacting with the institution's learning management system, as well as assessment mark files provided by the lecturer in the form of various comma delimited (CSV) files. The raw data for each of these sources can be found in Appendix A. This data was pre-processed in preparation for use by the selected machine learning techniques.

4.3 Data Pre-Processing

As mentioned in Section 3.2, data pre-processing is one crucial step in LA and EDM. The four main steps are represented in Figure 4.2. These steps include data cleaning, data integration, data selection and data transformation.

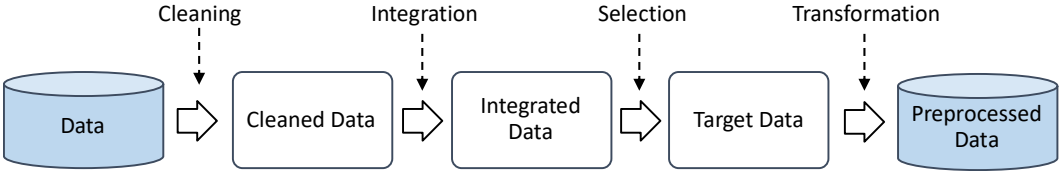

Figure 4.2: Data Pre-processing Steps

Each of these aspects is discussed in relation to the study in this section. The data attribute selection was performed and is motivated throughout the other phases of pre-processing (Section 4.3.1 – Section 4.3.4). The raw data contains personally identifiable data, which is required in the integration of the data; therefore, encoding in the form of pseudonyms is used whenever data sources are displayed. Before pre-processing can occur, the data must first be collected.

4.3.1 Data Collection

As mentioned in Section 4.2, the data used by the models were obtained from the third-year capstone project module (ONT3660) at the Nelson Mandela University from 2016 and 2017 and combined into a single dataset.

The structure and content of the Moodle action log files are displayed in Table 4.1.

Table 4.1: Action Log Structure and Contents

Column Name	Description
Date	The date the event occurred.
Time	The time at which the event occurred.
User full name	The name of the user who executed the event.
Affected user	The name of the user who was affected by the event.
Event context	A description of what the event was executed on.

Component	The component type event within the LMS.
Event name	The type of event that was executed.
Description	A description of the event that was executed.
Origin	The type of system from which the event originated.
IP address	The IP address of the event origin.

Figure 4.3 shows an example, with anonymised student names, of actual action log file data.

Date	Time	User Full Name	Affected User	Event Context	Component	Event Name	Description	Origin	IP Address
2017/06/02	16:06	John Doe	John Doe	Assignment: UPLOAD Sample Reports	File submissions	A file has been uploaded.	The user with id '22639' has uploaded a file to the submission with id '733922' in the assignment activity with course module id '142377'.	web	197.12.123.12
2017/06/04	14:03	Eric Smith	Eric Smith	File: Project PROGRESS Report Template	File	Course module viewed	The user with id '33085' viewed the 'resource' activity with course module id '143312'.	web	197.12.123.12

Figure 4.3: Action Log File Data

The layout and data contained in the assessment mark files provided by the lecturer varied slightly from year to year. However, they all contain the data required for this study as shown in Table 4.2.

Table 4.2: Mark Sheet Structure and Content

Attribute	Description
Student surname	The surname of the student.
Student name	The first name of the student.
Group name	The name of the group that the student is part of.
Iteration 1 mark	The mark obtained in the first quarter of the year.
Iteration 2 mark	The mark obtained in the second quarter of the year.

Iteration 3 mark	The mark obtained in the third quarter of the year.
Iteration 4 mark	The mark obtained in the final quarter of the year.
Final mark	The final project mark obtained.

Once the data from the Moodle log files and assessment mark files were collected, pre-processing could begin, starting with data cleaning.

4.3.2 Data Cleaning

Section 3.2.1 describes data cleaning as detecting and removing errors and inconsistencies in order to improve the quality of the data. The Moodle log files, and assessment mark files are discussed separately in terms of cleaning and can be found in Appendix A.

4.3.2.1 Data Cleaning of Moodle Log Files

In this step, the raw log files were processed and cleaned to prepare them for future processing. This step is important as the raw log files can have missing values, noisy data or redundant information. The raw log files contain up to 39000 rows of data; therefore, doing this manually is not possible. A C# application was developed to sanitise the Moodle log files automatically and prepare them for future steps.

The first part of this step is to remove noisy data, missing values and redundant information. The redundant information is identified as data that will not be necessary in building the dataset. This data was identified based on the component column of the logs. The values to be removed were *system, report, choice* and *forum* as they have no relation to student academic interactions in the context of this study. Listing 4.1 shows the code developed to perform this step.

```
1.  string[] Split = row.Split(';');
2.  if (Split.Where(x => string.IsNullOrWhiteSpace(x)).ToList().Count > 0) continue;
3.  if (Split[5] == "System" || Split[5].Contains("report") || Split[5] == "Choice"
4.                  || Split[5] == "Forum") continue;
```

Listing 4.1: Code to Remove Redundant Data

To allow the user events to be grouped and more easily identified by using the ID instead of name, the *description* column needed to be split. This resulted in obtaining the *user ID* and *event context ID* in separate columns. Redundant columns such as *affected user, origin* and *IP address* were also removed by only storing the columns required for future processing. Listing 4.2 shows the code developed to split the data and store relevant columns.

```
1.  Regex regex = new Regex("'(.*?)'");
2.  Match match = regex.Match(Split[7]);
3.  List<int> numbers = new List<int>();
4.  while (match.Success)
5.  {
6.      string value = match.Groups[1].Value;
7.
8.      if (int.TryParse(value, out int result))
9.      {
10.         numbers.Add(result);
11.     }
12.
13.     match = match.NextMatch();
14. }
15. if (Split[6] == "A submission has been submitted." || Split[6] == "Submission created."
16.                 || Split[6] == "A file has been uploaded.")
17. {
18.     one = numbers[0];
19.     two = numbers[2];
20. }
21. else
22. {
23.     one = numbers[0];
24.     two = numbers[1];
25. }
26.
```

```
27. Current = new Log()
28. {
29.
30.     Date = Split[0] + " " + Split[1],
31.     Name = Split[2],
32.     CourseModule = Split[4],
33.     Type = Split[5],
34.     Action = Split[6],
35.     UserID = one.ToString(),
36.     CourseModuleID = two.ToString()
37. };
38.
39. Logs.Add(Current);
```

Listing 4.2: Code to Format 'Description' column

The output of the data cleaning step was CSV files for each year containing the cleaned data from the raw Moodle log files. This log file now contained no missing values, noisy data nor redundant information and was ready for further processing. The personally identifiable data such as name and surname could not be removed at this stage as they were required in the data integration step. An extract of the anonymised content of this cleaned Moodle log file is displayed in Figure 4.4.

Date	User Full Name	Event Context	Component	Event Name	User ID	Context ID
2017/06/02	John Doe	Assignment: UPLOAD Sample Reports	File submissions	A file has been uploaded.	22639	142377
2017/06/04	Eric Smith	File: Project PROGRESS Report Template	File	Course module viewed	33085	143312

Figure 4.4: Extract of Cleaned Moodle Log File

4.3.2.2 Data Cleaning of Assessment Mark Files

The assessment mark files varied slightly from year to year. These variances included what data was stored and how that data was stored. It was necessary to perform the data cleaning manually on this data set. This was also possible as there were a manageable number of rows. The processing performed on this raw data

included normalising the columns, linking students to groups and removing noisy or missing data

Normalising the columns

This process involved transforming each year's data into the same format in terms of columns; in summary, ensuring that each year contained the same columns in the same order.

Linking students to groups and results

This process was only necessary as the student names and results were stored separately from the groups. The data containing student names and their associated groups were stored in one sheet, while the student names and iteration marks were stored in another.

Removing noisy or missing data

In this process, any students who had dropped out or changed groups before a specific point were removed or updated to a new group. Also, any students with no results stored were removed.

The result of this data cleaning were CSV files for each year containing the cleaned data from the raw assessment mark files. These files then contained no missing values, noisy data or redundant information, and they were ready for further processing. Similarly, as with the Moodle log files, the personally identifiable data,

namely student name and surname could not yet be removed. The newly normalised columns were as follows:

- Student surname
- Student name
- Iteration 1 mark
- Iteration 2 mark
- Iteration 3 mark
- Iteration 4 mark (Not used in final dataset)
- Final mark

Once the data was cleaned and processed, as mentioned above, the data was ready for the next phase of data pre-processing. This phase is the data integration phase and deals with combining the two data sources into a single file. These two data sources were the Moodle log files and the assessment mark files. The cleaned data within each of these data sources were then ready for integration.

4.3.3 Data Integration

As discussed in Section 3.2.4, data integration involves integrating and merging the data from multiple data sources. In this case, it refers to the merging of the Moodle log files, and the assessment mark files. Han, et al., (2012) mention several problems faced during data integration. One of these problems is the entity identification problem. In this study a problem that was faced was that there were no code or ID entities to connect the data for each student in the log files to the

corresponding student in the assessment mark files. Therefore, the only means to integrate these two data stores was based on the student name. However, this caused problems that will be discussed later in this section.

The first step of data integration is to create a new file with a single row for each unique student in the log files and to total up the relevant data for each event relating to said student. Each student is uniquely identified by the *user ID* column in the cleaned log files. The data to be stored was firstly, the number of unique files each student viewed throughout the year; and secondly, the number of assignments each student submitted throughout the year. Each assignment and file event was uniquely identified by the *context ID* column in the cleaned log file and associated with a student based on the *user ID* column.

Listing 4.3 shows the code developed to create a single student object for each unique student found in the log files

```
1.  bool InList;
2.  var count = Logs.Where(x => x.Type == "File").Select(x => x.CourseModuleID).Distinct()
    .Count();
3.  foreach (Log log in Logs)
4.  {
5.      InList = false;
6.
7.      //CHECKS IF STUDENT HAS ALREADY BEEN ADDED
8.      foreach (Student student in Students)
9.      {
10.         if (log.UserID == student.ID)
11.             InList = true;
12.     }
13.
14.     if (!InList)
15.     {
16.         Student = new Student()
17.         {
18.             ID = log.UserID,
19.             Name = log.Name
20.         };
21.         Students.Add(Student);
22.     }
23. }
```

Listing 4.3: Code to Create a Single Student Object for Each Unique Student

Listing 4.4 shows the code developed to calculate the total unique files that each student views. It is likely that each student would view the same file multiple times. Therefore, only the first instance that a student views a file was considered.

```
1.  bool IsIn;
2.  foreach (Student student in Students)
3.  {
4.      FileView FV;
5.      List<FileView> Files = new List<FileView>();
6.      foreach (Log log in Logs)
7.      {
8.          IsIn = false;
9.          if (log.Type == "File" && student.Name == log.Name)
10.         {
11.             if (Files != null)
12.             {
13.                 foreach (FileView fv in Files)
14.                 {
15.                     //CHECKS IF FILE HAS ALREADY BEEN VIEWED BY STUDENT
16.                     if (log.CourseModuleID == fv.FileID)
17.                     {
18.                         IsIn = true;
19.                     }
20.                 }
21.             }
22.
23.             //IF IT IS NOT, ADD IT TO THE LIST
24.             if (!IsIn)
25.             {
26.                          FV = new FileView()
27.                          {
28.                              Date = DateTime.Parse(log.Date),
29.                     UserID = log.UserID,
30.                     FileID = log.CourseModuleID
31.                 };
32.                 Files.Add(FV);
33.             }
34.         }
35.     }
36.     student.FilesViewed = Files;
37. }
```

Listing 4.4: Code to Store Unique File Views for Each Student

Listing 4.5 shows the code developed to calculate the number of assignments each student submitted. Some students may have uploaded the same assignment multiple times; therefore, only the first instance was considered.

```
1.  foreach (Student student in Students)
2.  {
3.      Upload UL;
4.      List<Upload> Uploads = new List<Upload>();
5.      foreach (Log log in Logs)
6.      {
7.          IsIn = false;
8.          if (log.Action == "A file has been uploaded." && student.Name == log.Name)
9.          {
10.             if (Uploads != null)
11.             {
12.                 foreach (Upload ul in Uploads)
13.                 {
14.                     //CHECKS IF ASSIGNMENT HAS ALREADY BEEN SUBMITTED BY STUDENT
15.                     if (log.CourseModuleID == ul.AssignmentID)
16.                     {
17.                         IsIn = true;
18.                     }
19.                 }
20.             }
21.
22.             //IF IT IS NOT, ADD IT TO THE LIST
23.             if (!IsIn)
24.             {
25.                 UL = new Upload()
26.                 {
27.                     Date = DateTime.Parse(log.Date),
28.                     UserID = log.UserID,
29.                     AssignmentID = log.CourseModuleID
30.                 };
31.                 Uploads.Add(UL);
32.             }
33.         }
34.     }
35.     student.AssignUpload = Uploads;
36. }
```

Listing 4.5: Code to Store Unique Assignment Uploads for Each Student

The next step was to integrate the assessment mark file data with the Moodle log file data. Normally, a unique identifier would be used to find the corresponding student in each data source. However, each source uses a different entity to uniquely identify students. The Moodle log files use the *user ID* column to uniquely identify students. In contrast, the assessment mark files use the *Student number* column. The *user ID* and *Student number* columns do not share the same values and, therefore,

could not be used to integrate the datasets. As a result, the only method to link each student in the Moodle log file, with the corresponding row in the assessment mark files, was the student's name. The code in Listing 4.6 performs this linking.

```
1.  foreach (Student student in Students)
2.  {
3.      foreach (MarkLine Line in Lines)
4.      {
5.                  //COMPARES NAME IN LOG TO NAME IN STUDENT (FROM MARK SHEET)
6.          if (Line.Name.ToLower() == student.Name.ToLower())
7.          {
8.              student.Group = Line.Group;
9.              student.IT1 = Line.IT1;
10.             student.IT2 = Line.IT2;
11.             student.IT3 = Line.IT3;
12.             student.IT4 = Line.IT4;
13.             student.Final = Line.Final;
14.             //COUNTS NUMBER OF FILEVIEW OBJECTS IN THE LIST
15.             student.NumberOfViews = student.FilesViewed.Count;
16.             student.NumberOfUploads = student.AssignUpload.Count;
17.         }
18.     }
19. }
```

Listing 4.6: Code to integrate Log File and Mark Sheet data

Once the assessment mark file data had been integrated into the log file data, the output was a file containing one row for each student. The resulting file contained the following data:

- User ID
- User name
- Group name
- Iteration 1 mark
- Iteration 2 mark
- Iteration 3 mark
- Iteration 4 mark
- Final mark

- Number of unique files viewed
- Number of assignments uploaded

One of the issues faced in data integration is the entity identification problem. In this study, student names were used as the linking entities between data stores. This is an issue, as in many situations, mostly owing to human error, so the names are stored differently in each of the data stores.

For example, in the Moodle log files the names were stored in full such as 'Ryan Vaughan Evezard', whereas, in the assessment mark files, they were stored as 'EVEZARD, RV' and 'Ryan' in a separate column. Using the previous example, the names would link correctly; however, problems arose when the names are misspelt in either of the data stores or in some situations when the second name was stored as the Student name entity. This problem was solved by manually changing the names of the students in the assessment mark files to match the names in the log files where the entities did not match.

The final step of data integration was to *combine the students into groups*. This was achieved by using the *Group name* entity to identify which students belonged to the same group. The *Iteration mark* columns are the same for each student in the group and therefore did not need to be modified. The *Files viewed* and *Assignment uploads* entities, however, were different for each student as only one student per group had to upload weekly deliverables. These entities were totalled to get a value for the entire group. The member who viewed the lowest number of files per group, together with the member with the highest assignment uploads per group, were

calculated and stored. Finally, the number of group members were totalled and stored. The code shown in Listing 4.7 represents the above grouping process.

```
1.  foreach (Student student in Students)
2.  {
3.      IsIn = false;
4.
5.      if (Groups != null)
6.      {
7.          foreach (Group group in Groups)
8.          {
9.              if (student.Group == group.Name)
10.             {
11.                 IsIn = true;
12.                 group.Members++;
13.                 group.TotalFileViews += student.NumberOfViews;
14.                 group.TotalUplaods += student.NumberOfUploads;
15.                 if (student.NumberOfUploads > group.HighestUploads)
16.                     group.HighestUploads = student.NumberOfUploads;
17.                 if (student.NumberOfViews < group.LowestViews)
18.                     group.LowestViews = student.NumberOfViews;
19.             }
20.         }
21.     }
22.
23.     if (!IsIn)
24.     {
25.         gr = new Group()
26.         {
27.             ID = Count.ToString(),
28.             Name = student.Group,
29.             IT1 = student.IT1,
30.             IT2 = student.IT2,
31.             IT3 = student.IT3,
32.             IT4 = student.IT4,
33.             Final = student.Final,
34.             TotalUplaods = student.NumberOfUploads,
35.             TotalFileViews = student.NumberOfViews,
36.             LowestViews = student.NumberOfViews,
37.             HighestUploads = student.NumberOfUploads
38.         };
39.         Groups.Add(gr);
40.         Count++;
41.     }
42. }
```

Listing 4.7: Code to Group Student Data Based on Group Name

Once all the above pre-processing had been applied, the resulting dataset was saved as a CSV file. This file contained cleaned, integrated and redundant free data to be used in the modelling process. However, before modelling could be done on the data, one additional pre-processing phase was required. This was the data transformation phase. Figures 4.5, 4.6, and 4.7 show the dataset for each iteration

respectively. Once the data was integrated into a single file, the data had to be transformed into a format required by the modelling process.

Group Members	IT 1	Final	File Views	Uploads	AVG File Views	Lowest Views	AVG Uploads	Highest Uploads
3	59	50	53	3	17	12	1	3
4	47	60	42	3	10	4	0	1
4	65	40	64	3	16	8	0	3

Figure 4.5: Extract of Dataset for Quarter 1

Group Members	IT 1	IT 2	Final	File Views	Uploads	AVG File Views	Lowest Views	AVG Uploads	Highest Uploads
3	59	46	50	75	10	25	19	3	9
4	47	43	60	71	11	17	12	2	3
4	65	38	40	102	10	25	11	2	10

Figure 4.6: Extract of Dataset for Quarter 2

Group Members	IT 1	IT 2	IT 3	Final	File Views	Uploads	AVG File Views	Lowest Views	AVG Uploads	Highest Uploads
3	59	46	50	50	91	17	30	23	5	15
4	47	43	44	60	88	17	22	15	4	6
4	65	38	36	40	119	17	29	11	4	13

Figure 4.7: Extract of Dataset for Quarter 3

The previous sections performed the data pre-processing required for processing the raw data into a format that can be used as an input dataset for prediction models. However, data transformation is required as a final phase so that the prediction models can make use of the dataset more effectively.

4.3.4 Data Transformation: Normalisation

Data transformation is defined by Uma and Hanumanthappa (2017) as the converting of source data into the format required by the modelling process. This phase is completed so that the modelling process may be done more efficiently and accurately. For this study, normalisation was necessary, and was applied to the dataset in the form of feature scaling.

Feature scaling is required as the range of values in the raw data vary widely. For example, in this study the values of the ***File views*** feature were considerably larger than those in the *Group members* feature. This is a problem as most machine learning models use the Euclidian distance between two data points. As a result, features with a higher magnitude will outweigh the features with lower magnitudes in the resulting prediction models (Ioffe & Szegedy, 2015).

Another common normalisation step is the encoding of categorical data. This step was not required in this study as the dataset does not contain any categorical data. The data at this stage was completely pre-processed and ready for use in the modelling process.

4.3.5 Pre-Processed Data

Once all the above pre-processing steps had been applied, the resulting dataset was ready for modelling and saved as a CSV file. This data was integrated, cleaned, anonymised and enriched, based on the processes discussed in Section 3. This data no longer contained personally identifiable data and therefore, in this state, could

be shared publicly for reproduction. Figure 4.8 shows an extract of the pre-processed data set. The pre-processed data in this form was ready to be used in the modelling process.

Group Members	IT 1	IT 2	IT 3	File Views	Uploads	AVG File Views	Lowest Views	AVG Uploads	Highest Uploads	Final
0.449712	-0.640511	-2.19896	-0.895975	1.14707	0.196315	1.32513	0.430691	-0.478709	-0.064215	0
-0.426043	-1.01314	-1.80712	-1.21744	-0.425518	-0.464018	-0.192234	-0.825315	-0.109704	-0.623263	0
-0.426043	-0.118837	-1.07943	-1.29781	-0.0597999	-1.12435	0.756119	0.11669	-0.847714	-1.18231	0

Figure 4.8: Extract from Pre-Processed Dataset

With the data normalised, the resulting dataset is completely pre-processed into a format that will allow for the effective and accurate creation of prediction models.

4.4 Prediction Modelling

Chapter 3 discussed three different machine learning techniques, namely, artificial neural networks, decision trees and naïve Bayes classifiers. Each of these techniques was used in this study. While Section 3.4 proposed an artificial neural network, Section 3.5 proposed a decision tree making use of the C4.5 algorithm, and Section 3.6 proposed a naïve Bayes classifier. This section describes the implementation of each of these machine learning techniques in creating prediction models. The full code for each of the models can be found in Appendix B.

4.4.1 Prediction Model Preparation

As mentioned in Section 4.1, one of the purposes of this study was to compare the results and performance of the prediction models based on the selected machine learning techniques mentioned above. In order to do this, the data provided to the prediction models must be consistent. The data was split into training and test sets. A split of 50% for training data and 50% for testing data was decided upon. Using 50% of the data for testing ensured that the resulting predictions were an accurate representation of the prediction model on unseen data. This split also allowed for more consistent results across prediction models.

K-fold cross-validation was used to test the performance of each of the prediction models. This technique is used to solve the variance problem when testing a model. In summary, k-fold cross-validation splits the training set into 10 folds. The model is trained on 9 folds and then tested on the last remaining fold. This gives a better indication of the model performance as an average of the different accuracies is taken, as well as the standard deviation, which indicates the variance.

In addition to accuracy, the metrics presented in Table 4.3 were used to evaluate the performance of the prediction models:

Table 4.3: Equations for Various Performance Metrics

Metric	Equation
Precision	$Precision = \frac{TP}{TP + FP}$
Recall	$Recall = \frac{TP}{TP + FN}$

F-Measure	$F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$

These metrics are discussed in detail in Section 3.7, and obtain their values from confusion matrices.

Each of the prediction models was modelled, based on three separate datasets. These datasets were labelled: Quarter 1, Quarter 2 and Quarter 3. Each prediction model was evaluated and compared based on each of these datasets separately.

4.4.2 Prediction Model Execution

The prediction modelling process for each of the three models is summarised in the following sections. The code used to create each of the prediction models can be found in Appendix B.

4.4.2.1 Artificial Neural Network Execution

In order to create the artificial neural network prediction model, the Keras python library was used to fit the classifier to the dataset. This library allows for the modelling of an artificial neural network where the number of inputs, hidden layers and parameters can be altered. Parameters for the artificial neural network, such as the activation functions, were selected and motivated in Section 3.5, namely the rectifier activation function for all hidden layers, and the sigmoid activation function for the output layer. The dropout package within the Keras library allows for random features to be removed during training in order to prevent the model overfitting to the training data. Overfitting occurs when a model memorises and becomes overfit to the training data; therefore, performing poorly when tested. The

performance of the artificial neural network prediction model is presented in Section 5.3.1.

4.4.2.2 Decision Tree Execution

In order to create the decision tree prediction model, the DecisionTreeClassifier package was used from the scikit-learn library to fit the classifier to the dataset. This library uses an optimised version of the classification and regression tree (CART) algorithm, which is similar to C4.5. However, CART supports numerical target values, whereas C4.5 does not. This algorithm uses the Gini index as the attribute selection method. This information gain algorithm was discussed in detail in Section 3.6. The performance of the decision tree prediction model is presented in Section 5.3.2

4.4.2.3 Naïve Bayes Execution

In order to create the naïve Bayes prediction model, the GaussianNB package was used from the scikit-learn library to fit the classifier to the dataset. This package uses a Gaussian naïve Bayes classifier, which is motivated in Section 3.7. This library also contains packages for multinomial and Bernoulli naïve Bayes classifiers; however, as mentioned in Section 3.7, the GaussianNB was used in this study. The performance of the naïve Bayes prediction model is presented in Section 5.3.3.

4.5 Conclusion

This chapter presented the design and details of the experiment conducted in this study. Firstly, data was gathered and pre-processed to be used as input by the selected machine learning techniques. A dataset was created for the first three quarters of the year. Each of the selected techniques was then trained and tested on each of these datasets.

The first step of exploratory experimentation is experiment design. The main focus of this chapter was designing the experiment. This involved selecting and pre-processing the data and preparing the prediction models. The next step of exploratory experimentation is to conduct the experiment. Section 4.4.2 provides a description of how the experiment was conducted. In designing and conducting this experiment, prediction models were created using the selected machine learning models from Chapter 3 which address SO_4: To use the selected machine learning techniques to create prediction models to predict student group academic performance.

This experimentation resulted in numerous evaluation metrics, namely accuracy, f-measure, and variance, for each of the prediction models. The metrics obtained in this chapter are listed and analysed individually and comparatively in Chapter 5.

Chapter 5: Prediction Model Analysis

5.1 Introduction

The previous chapter described the steps taken to obtain the final dataset, as well as how the prediction models were trained and created. In this research three different prediction models were created, namely artificial neural networks (Section 4.3.2.1), decision trees (Section 4.3.2.2), and naïve Bayes classifiers (Section 4.3.2.3). The results produced by the various prediction models are provided and analysed in this chapter. The layout and research objective of this chapter are illustrated in Figure 5.1. This chapter addresses the following research objective:

SO_5**:** *To evaluate and compare the performance of selected machine learning techniques in predicting student group academic performance.*

Before providing the results of each model, it is important to understand the context in which these results are given. The distribution of the final datasets provides insight into the quality of the dataset (Section 5.2). The results of each of the three prediction models, neural network model (Section 5.3.1), decision tree model (Section 5.3.2) and naïve Bayes model (Section 5.3.3) are presented and discussed. A comparison of the respective results for each prediction model makes visualising the differences between each model easier (Section 5.4)

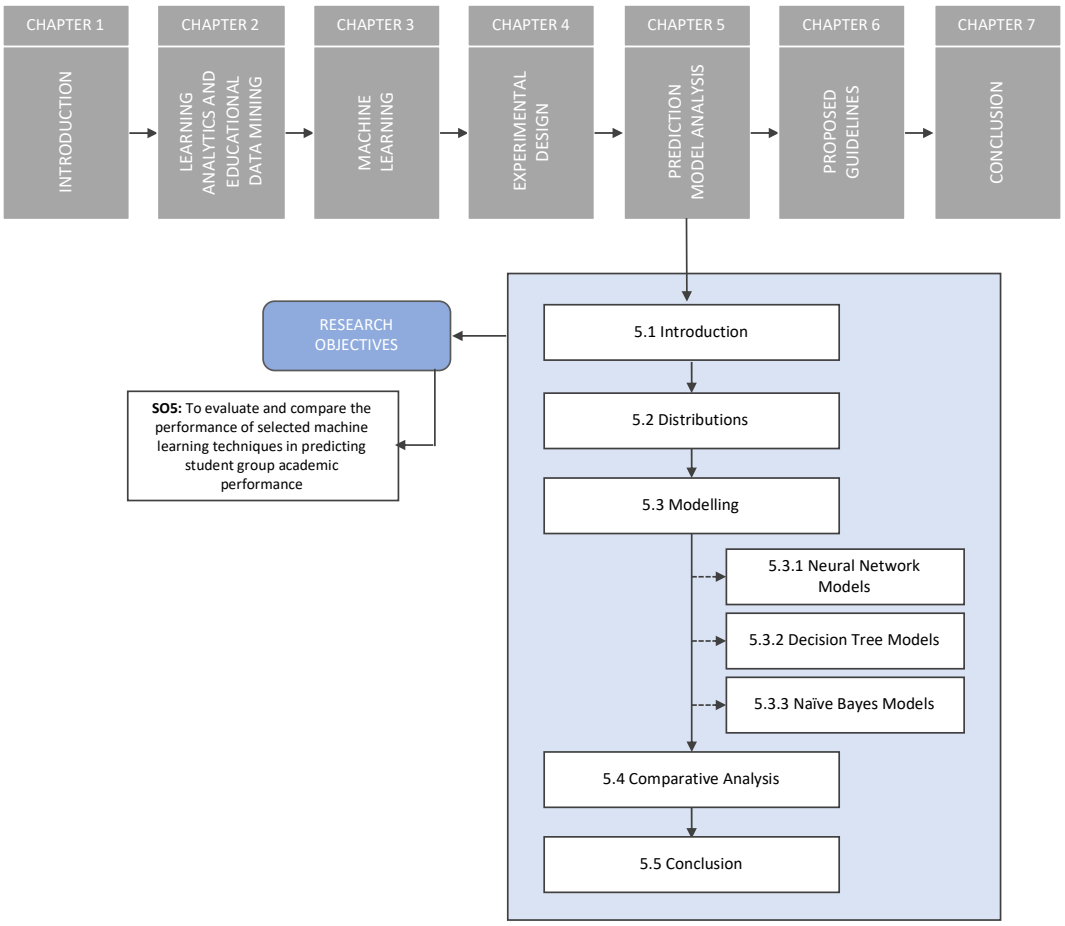

Figure 5.1: Chapter Five Outline

5.2 Target Distribution

To gain an understanding of the quality of the final dataset, the distribution of the target data should be examined. In this study, the target data contained two classes, *pass* and *fail*. A perfectly balanced and, therefore, optimal dataset would have 50% entries for each of the classes. However, the target dataset contained 79 group *pass* entries and 27 group *fail* entries. This resulted in a distribution of 75% to 25%.

As discussed in Section 3.8, this distribution is considered imbalanced and, therefore, has some problems associated with it. When dealing with an imbalanced distribution, the *accuracy* performance metric is not an accurate measure of a

model's performance. Therefore, when comparing models, the metric's precision, recall, and f-measure were used as discussed in Section 3.8.

5.3 Prediction Models

The results of each of the three prediction models are provided in this section. Each model includes results for the three datasets, namely Quarter 1, Quarter 2 and Quarter 3. For each of these datasets, the resulting confusion matrix is included, along with the metrics mentioned in Section 5.2. Table 5.1 shows an example of the confusion matrix which is explained in Section 3.8.

Table 5.1 Confusion Matrix

	Predicted Fail (0)	Predicted Pass (1)
Actual Fail (0)	True Positive	False Negative
Actual Pass (1)	False Positive	True Negative

The first prediction model results to be analysed are from the artificial neural network models.

5.3.1 Artificial Neural Networks

Quarter 1

Table 5.2 shows a confusion matrix of the results after evaluation on the ANN model described in Section 4.4.2.1. These results are based on the Quarter 1 dataset. The ANN model correctly predicted 8 fails out of 17 and 32 passes out of 36.

Table 5.2: Confusion Matrix for ANN Quarter 1

	Predicted Fail (0)	Predicted Pass (1)
Actual Fail (0)	8	9
Actual Pass (1)	4	32

Table 5.3 is a summary of the performance of the ANN model. The metrics are derived from the values in the corresponding confusion matrix as shown in Table 5.2. The ANN model obtained an accuracy of 75% and an f-measure of 55%.

Table 5.3: Performance Metrics for ANN Quarter 1

Metric	**Mean**	**Variance**
Accuracy	75%	0.057
Precision	67%	
Recall	47%	
F-Measure	55%	

Quarter 2

Table 5.4 shows a confusion matrix of the results after evaluation on the ANN model described in Section 4.4.2.1. These results are based on the Quarter 2 dataset. The ANN model correctly predicted 10 fails out of 17 and 35 passes out of 36.

Table 5.4: Confusion Matrix for ANN Quarter 2

	Predicted Fail (0)	Predicted Pass (1)
Actual Fail (0)	10	7
Actual Pass (1)	1	35

Table 5.5 is a summary of the performance of the ANN model. The metrics are derived from the values in the corresponding confusion matrix Table 5.4. This model obtained an accuracy of 85% and an f-measure of 71%.

Table 5.5: Performance Metrics for ANN Quarter 2

Metric	Mean	Variance
Accuracy	85%	0.048
Precision	91%	
Recall	59%	
F-Measure	71%	

Quarter 3

Table 5.6 shows a confusion matrix of the results after evaluation on the ANN model described in Section 4.4.2.1. These results are based on the Quarter 3 dataset. The ANN model correctly predicted 13 fails out of 15 and 35 passes out of 38.

Table 5.6: Confusion Matrix for ANN Quarter 3

	Predicted Fail (0)	Predicted Pass (1)
Actual Fail (0)	13	2
Actual Pass (1)	3	35

Table 5.7 is a summary of the performance of the ANN model. The metrics are derived from the values in the corresponding confusion matrix as shown in Table 5.6. The ANN model obtained an accuracy of 91% and an f-measure of 84%.

Table 5.7: Performance Metrics for ANN Quarter 3

Metric	Mean	Variance
Accuracy	91%	0.036

Precision	81%	
Recall	87%	
F-Measure	84%	

The next prediction model results to be analysed are from the decision tree models.

5.3.2 Decision Trees

Quarter 1

Table 5.8 shows a confusion matrix of the results after evaluation on the decision tree model described in Section 4.4.2.2. These results are based on the Quarter 1 dataset. This decision tree model correctly predicted 3 fails out of 10 and 38 passes out of 43.

Table 5.8: Confusion Matrix for DT Quarter 1

	Predicted Fail (0)	Predicted Pass (1)
Actual Fail (0)	3	7
Actual Pass (1)	5	38

Table 5.9 is a summary of the performance of the decision tree model. The metrics are derived from the values in the corresponding confusion matrix as shown in Table 5.8. This decision tree model obtained an accuracy of 77% and an f-measure of 33%.

Table 5.9: Performance Metrics for DT Quarter 1

Metric	Mean	Variance
Accuracy	77%	0.033
Precision	38%	

Recall	30%	
F-Measure	33%	

Quarter 2

Table 5.10 shows a confusion matrix of the results after evaluation on the decision tree model described in Section 4.4.2.2. These results are based on the Quarter 2 dataset. This decision tree model correctly predicted 14 fails out of 15 and 35 passes out of 38.

Table 5.10: Confusion Matrix for DT Quarter 2

	Predicted Fail (0)	Predicted Pass (1)
Actual Fail (0)	14	1
Actual Pass (1)	3	35

Table 5.11 is a summary of the performance of the decision tree model. The metrics are derived from the values in the corresponding confusion matrix as shown in Table 5.10. This decision tree model obtained an accuracy of 92% and an f-measure of 88%.

Table 5.11: Performance Metrics for DT Quarter 2

Metric	**Mean**	**Variance**
Accuracy	92%	0.024
Precision	82%	
Recall	93%	
F-Measure	88%	

Quarter 3

Table 5.12 shows a confusion matrix of the results after evaluation on the decision tree model described in Section 4.4.2.2. These results are based on the Quarter 3 dataset. This decision tree model correctly predicted 13 fails out of 14 and 38 passes out of 39.

Table 5.12: Confusion Matrix for DT Quarter 3

	Predicted Fail (0)	Predicted Pass (1)
Actual Fail (0)	13	1
Actual Pass (1)	1	38

Table 5.13 is a summary of the performance of the decision tree model. The metrics are derived from the values in the corresponding confusion matrix as shown in Table 5.12. This decision tree model obtained an accuracy of 96% and an f-measure of 93%.

Table 5.13: Performance Metrics for DT Quarter 3

Metric	**Mean**	**Variance**
Accuracy	96%	0.019
Precision	93%	
Recall	93%	
F-Measure	93%	

The next prediction model results to be analysed are from the naïve Bayes classifier models.

5.3.3 Naïve Bayes Classifier

Quarter 1

Table 5.14 shows a confusion matrix of the results after evaluation on the naïve Bayes classifier model described in Section 4.4.2.3. These results are based on the Quarter 1 dataset. This naïve Bayes classifier model correctly predicted 5 fails out of 12 and 34 passes out of 41.

Table 5.14: Confusion Matrix for NB Quarter 1

	Predicted Fail (0)	Predicted Pass (1)
Actual Fail (0)	5	7
Actual Pass (1)	7	34

Table 5.15 is a summary of the performance of the naïve Bayes classifier model. The metrics are derived from the values in the corresponding confusion matrix as shown in Table 5.14. This naïve Bayes classifier model obtained an accuracy of 74% and an f-measure of 42%.

Table 5.15: Performance Metrics for NB Quarter 1

Metric	Mean	Variance
Accuracy	74%	0.074
Precision	42%	
Recall	42%	
F-Measure	42%	

Quarter 2

Table 5.16 shows a confusion matrix of the results after evaluation on the naïve Bayes classifier model described in Section 4.4.2.3. These results are based on the Quarter 2 dataset. This naïve Bayes classifier model correctly predicted 9 fails out of 15 and 35 passes out of 38.

Table 5.16: Confusion Matrix for NB Quarter 2

	Predicted Fail (0)	Predicted Pass (1)
Actual Fail (0)	9	6
Actual Pass (1)	3	35

Table 5.17 is a summary of the performance of the naïve Bayes classifier model. The metrics are derived from the values in the corresponding confusion matrix as shown in Table 5.16. This naïve Bayes classifier model obtained an accuracy of 83% and an f-measure of 67%.

Table 5.17: Performance Metrics for NB Quarter 2

Metric	Mean	Variance
Accuracy	83%	0.052
Precision	75%	
Recall	60%	
F-Measure	67%	

Quarter 3

Table 5.18 shows a confusion matrix of the results after evaluation on the naïve Bayes classifier model described in Section 4.4.2.3. These results are based on the

Quarter 3 dataset. This naïve Bayes classifier model correctly predicted 9 fails out of 10 and 41 passes out of 43.

Table 5.18: Confusion Matrix for NB Quarter 3

	Predicted Fail (0)	Predicted Pass (1)
Actual Fail (0)	9	1
Actual Pass (1)	2	41

Table 5.19 is a summary of the performance of the naïve Bayes classifier model. The metrics are derived from the values in the corresponding confusion matrix as shown in Table 5.18. This naïve Bayes classifier model obtained an accuracy of 94% and an f-measure of 86%.

Table 5.19: Performance Metrics for NB Quarter 3

Metric	Mean	Variance
Accuracy	94%	0.029
Precision	82%	
Recall	90%	
F-Measure	86%	

The performance of each of the prediction models having been analysed individually, the results can be summarised and analysed comparatively to gain a better understanding of how the models compare to one another.

5.4 Comparative Analysis

Having created the various models for each of the selected machine learning techniques trained on each of the three datasets and having gathered the resulting

performances, the differences between the respective results can be visualised. As mentioned in Section 3.8, accuracy is not a good measure of a model's performance as the dataset is imbalanced. F-measure represents a combination of the precision and recall metric. Therefore, the f-measure metric was used for this comparison. This comparison of f-measures can be seen in Figure 5.2. These comparative f-measures are discussed further in Chapter 6.

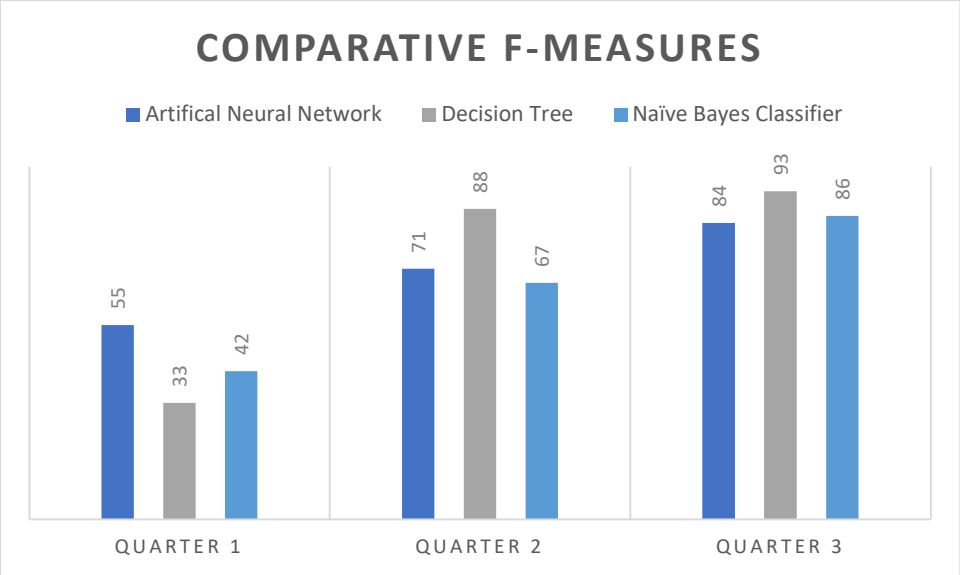

Figure 5.2: Comparative F-Measures

A second comparison is made between each model's variance. Variance indicates the stability of the testing accuracies versus would-be accuracies based on unseen data. Essentially, this measures how likely the model is to achieve the same performance on new data. This comparison of variances can be seen in Figure 5.3. These comparative variances are discussed further in Chapter 6.

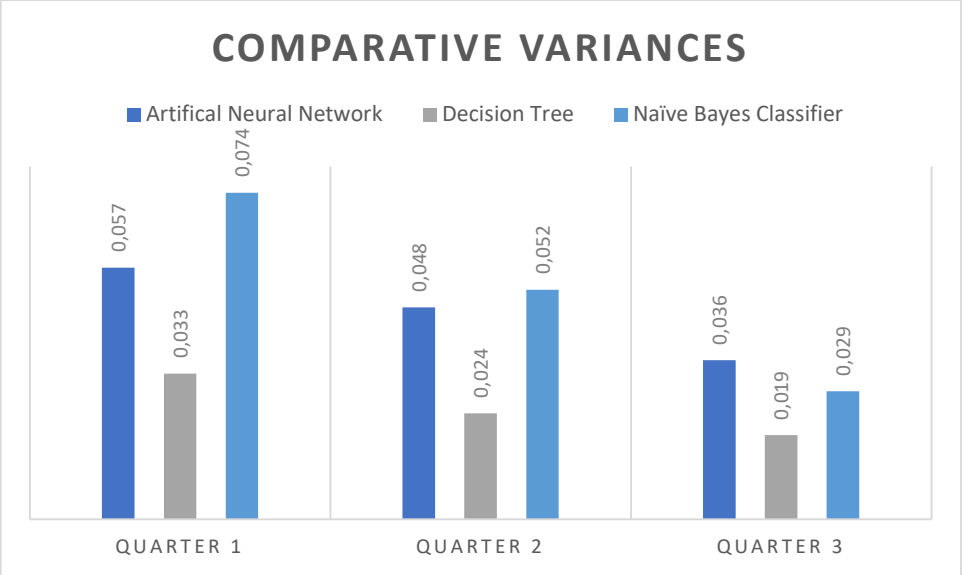

Figure 5.3: Comparative Variances

5.5 Conclusion

This chapter listed and analysed the results from the experimentation described in Chapter 4. The target distributions of the dataset were first analysed. Secondly, the results from each of the machine learning techniques, when trained on each of the datasets, was listed and analysed. Thus, SO_5 was addressed: *To evaluate and compare the performance of selected machine learning techniques in predicting student group academic performance.*

This analysis allows for an effective discussion on each of the machine learning techniques, individually and comparatively. The analysis of these results is discussed in detail in Chapter 6 together with a discussion of the literature in Chapter 2 and Chapter 3, as well as the experiment described in Chapter 4. Finally, Chapter 6 creates guidelines from the aforementioned discussions, in order to address the primary objective of this study.

Chapter 6: Proposed Guidelines

6.1 Introduction

The previous chapter presented and analysed the results from the experimentation described in Chapter 4. This chapter follows the steps of extracting knowledge from data to discuss the literature and findings of this study. From these discussions, guidelines are provided in order to address the primary objective of this study. The layout and research objectives of the chapter are illustrated in Figure 6.1. This chapter addresses the primary research objective of this study:

PO: *To provide guidelines for the use of machine learning techniques to predict academic performance of student project groups.*

This chapter follows the steps of extracting knowledge from data used in education data mining and learning analytics to discuss the literature and findings of this study. The discussion of the guidelines is divided into three sections. Firstly, the selection of the raw data (Section 6.2.1) is discussed, followed by detail relating to data pre-processing and transformation (Section 6.2.2). An individual and comparative discussion on each of the machine learning techniques and their respective prediction models follows (Section 6.2.3), with regard to the analysis performed in Chapter 5. Finally, a summary of the guidelines is provided (Section 6.3).

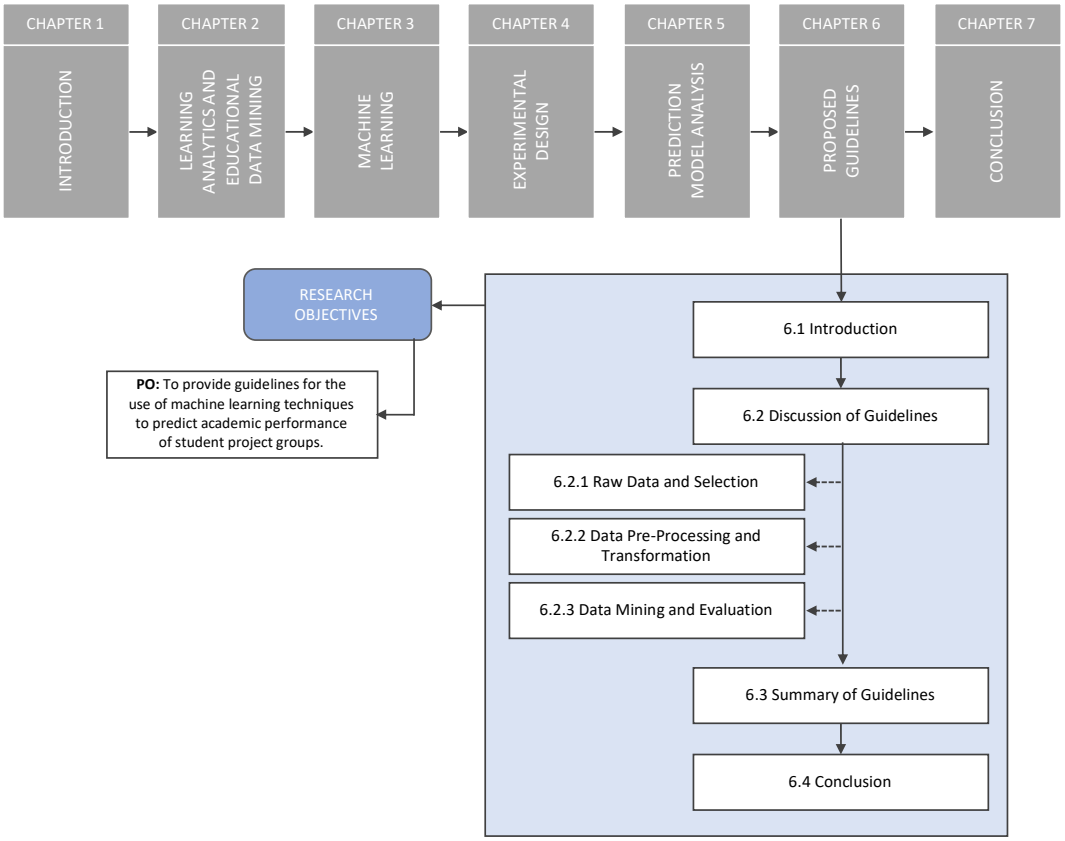

Figure 6.1: Chapter Outline

6.2 Discussion of Guidelines

The primary objective of this study was to provide guidelines for the use of machine learning techniques to predict academic performance of student project groups. This section discusses guidelines developed from observations made throughout the duration of this study. Chapter 1 discusses the steps of extracting knowledge from data as seen in Figure 6.2 (Baradwaj & Pal, 2012). Each of the guidelines is discussed in order and with regard to each of these steps.

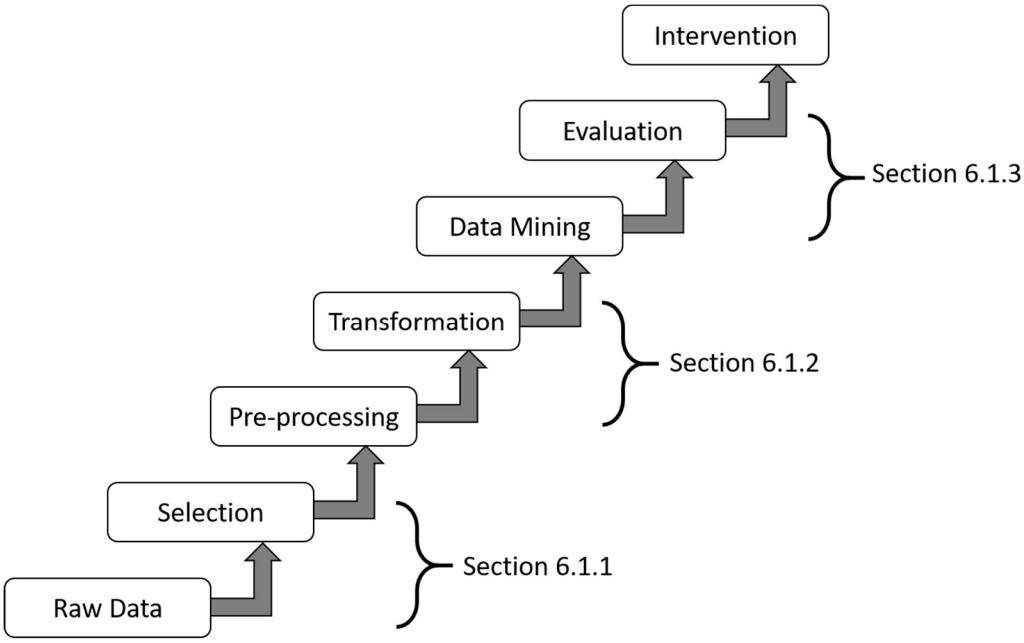

Figure 6.2: The Steps of Extracting Knowledge from Data (adapted from Baradwaj & Pal, 2012)

6.2.1 Raw Data and Selection

For this study, data was obtained from Moodle action log files together with corresponding assessment mark files for the third-year group project module, ONT3660 for 2016 and 2017. This data was extracted, as discussed in Chapter 3, and used to build the datasets to train and evaluate the various machine learning models. Chapter 3 presents the data selected from these data stores as:

- Iteration marks from the assessment mark files in CSV format.
- Final marks from the assessment mark files in CSV format.
- File view data from the Moodle action log files in CSV format.
- Assignment upload data Moodle action log files in CSV format.
- Number of group members from the assessment mark files in CSV format.

The most important and challenging aspect to consider when creating prediction models is the quality and quantity of the data that is available (Domingos, 2012).

Chapter 2 mentions that a rich dataset, along with numerous attributes, is the basis for advanced learning analytics and educational data mining. Therefore, the dataset used in this study can be improved by adding additional attributes and characteristics. Section 2.6.1 mentions attributes in existing research used to predict student academic performance. Attributes identified as being effective performance predictors that were not available for this study include:

- Student demographics
- Previous year results
- Lecture attendance
- External module results
- Secondary education results

Neither the Moodle log files, nor the assessment mark files stored the above-mentioned data and, therefore, were not available for this study. This data is currently stored in separate systems with the exception of lecture attendance. Lecture attendance would need to be tracked and stored throughout the year. In summary, modifications to existing data storage techniques would need to be made to cater for the collection of additional attributes. Therefore, the first guideline relating to this raw data and selection (Guideline 1.1) is:

Modify the existing data storage methods to allow for the collection of additional attributes when appropriate.

The target attribute for the dataset of this study is the final assessment. The final assessment is a binary categorical value where 1 represents a pass and 0 represents

a fail. The target distribution is 75% positive outcomes representing a pass and 25% negative outcomes representing a fail. This is an imbalanced distribution and, therefore, resulted in some problems, which were discussed in Section 5.2. A better distribution would be closer to 50% positive, 50% negative, and would yield better results in the modelling process.

This study made use of data from 2016 and 2017 which was combined into a single data set. Data from all available years should be used to increase the size of the dataset. With a higher quantity of data, the dataset can be modified to allow for a more balanced target distribution. Some modifications include (KDnuggets, 2019):

- *Under-sampling* - This method balances the dataset by reducing the size of the abundant class and requires a sufficient quantity of data.
- *Over-sampling* - This method is used when the quantity of data is insufficient. It balances the distribution by increasing the quantity of rare classes. One way that this is done is by duplicating some of the rare samples.
- *Cluster the abundant class* - This method splits the abundant class into sub-classes. For example, in the prediction of student academic performance, the pass outcome could be split into *pass* and *distinction*. This would result in three target classes with a more balanced distribution.

In machine learning a balanced target distribution will result in a more effective creation of models with better performance and lower variance. This leads to the second guideline relating to raw data and selection (Guideline 1.2):

Balance the input dataset through the use of appropriate techniques to prevent an unbalanced target distribution.

Once the raw data has been selected and collected, the next step of extracting knowledge from data is to pre-process and transform the collected data.

6.2.2 Data Pre-Processing and Transformation

A C# application was developed, as part of the study, to perform data preparation and pre-processing. This application was developed as the size of the log files was too large to process manually. The full process and code were discussed in full in Section 4.3.

Data cleaning and attribute selection were the first two steps of data pre-processing performed in this study. This step involved removing missing values and redundant information in the Moodle log files. The assessment mark files required manual normalisation of columns, linking students to their respective groups, and removing noisy and missing data.

Data integration was the next step performed. This step involved integrating the data from the log files together with the data from the assessment mark files. The entity identification problem was apparent owing to not having unique identifiers for students that corresponded across the two files. Therefore, student names were required to be used in this study to integrate the files. This resulted in the problem of needing to edit student names manually to match across data stores.

The final step of data pre-processing was data transformation. This step involved converting the source data into the format required by the modelling process. Normalisation was performed in the form of feature scaling. Feature scaling transformed all values into a certain range, so that features with higher magnitudes do not outweigh features with lower magnitudes.

As mentioned above, this study made use of a C# application to perform data preparation and pre-processing. This application was developed as the size of the Moodle log files were too large to process manually. The automation of this process proved effective in preparing and pre-processing the data efficiently. Therefore, it is recommended that a similar application be used. Hence, the first guideline relating to data pre-processing and transformation (Guideline 2.1) is:

Automate the preparation and pre-processing of the data by making use of an application.

When integrating the data, issues were faced relating to the entity identification problem. Student names were the only available entity for linking student data between the two files. To avoid this problem, a corresponding unique identifier should be used in both the log files and the assessment mark files. An example would be to store student numbers in both the log files and assessment mark files. This will allow for the complete automation of the data integration step. This leads to the second guideline for data pre-processing and transformation (Guideline 2.2):

Ensure that all data stores contain a corresponding unique identifier.

Assessment mark file data is currently stored inconsistently across years; however, it should be stored in a predetermined and consistent format similar to that discussed in Section 4.3.2.2. This will allow for the complete automation of the preparation and pre-processing of the data. This leads to the final guideline relating to data pre-processing and transformation (Guideline 2.3):

Store assessment mark file data in a consistent predetermined format.

Once the data had been pre-processed and transformed it was ready to be used by the modelling process, as discussed in Section 4.4.

6.2.3 Data Mining and Evaluation

In order to measure the performance of the prediction models, various performance metrics were used to allow for comparative analysis. These metrics included accuracy, precision, recall and f-measure.

Although accuracy was recorded, as it is the simplest metric to understand and a good baseline metric, it is not an effective measure of model performance. The accuracy of the prediction models was affected by the imbalanced distribution of the dataset and, therefore, does not show correctly how the minority class is classified. The minority class in this study were the negative outcomes.

Precision and recall were recorded so that the f-measure could be calculated. These equations and metrics were discussed in detail in Section 3.8. F-measure was the final evaluation criterion for the prediction models. F-measure was used as it takes both precision and recall into account. Together these metrics give an effective

representation of both successful positive predictions and successful negative predictions.

In summary, accuracy can be used as a good baseline metric for performance. However, it falls short in certain situations such as in an unbalanced dataset. Precision determines how accurately a model predicts positive outcomes as does recall with negative outcomes. A good model must have a good combination of successful positive predictions and successful negative predictions. The f-measure metric should be used for final evaluation as it combines both precision and recall into a single value. Therefore, the first guideline relating to data mining and evaluation (Guideline 3.1) is:

Evaluate the prediction models based on the f-measure as the final evaluation criterion.

Finally, the variance for each prediction model was recorded. The variance of a model represents the standard deviation of a model's performance after k-fold cross-validation. The variance of the prediction models indicates how likely it is to achieve the same performance on unseen data. The lower the variance of the prediction model, the more likely it is to achieve the same results. This leads to the second guideline for data mining and evaluation (Guideline 3.2):

Cross-validate using k-fold cross-validation when testing the prediction models and record the variance of the models.

There were three prediction models, each trained on three different datasets that were evaluated in this study. These prediction models were created based on the

machine learning techniques selected in Section 3.4. The selected techniques were artificial neural networks, decision trees and naïve Bayes classifiers.

6.2.3.1 Artificial Neural Networks

An artificial neural network model was created for each of the three datasets in Section 4.4. Artificial neural networks achieved an accuracy of 75% and an f-measure of 55% for the Quarter 1 dataset (Table 5.3). Table 5.2 indicates that the model correctly predicted only 8 of 17 negative outcomes. These results indicate that at this stage of the year, the model is mostly assuming a positive prediction. This is due to the imbalanced nature of the dataset. Therefore, an effective prediction cannot be made at this stage.

When trained on the Quarter 2 dataset, artificial neural networks yielded a higher performance with an accuracy of 85% and an f-measure of 71% (Table 5.5). Table 5.4 indicates that the model correctly predicted 10 of 17 negative outcomes. These results indicate that at this stage of the year, the model is still mostly assuming a positive prediction, although to a lesser degree. Therefore, a somewhat effective prediction can be made at this stage.

When trained on the Quarter 3 dataset, artificial neural networks showed a significant increase in both accuracy and f-measure when compared to the previous two datasets. The artificial neural network model achieved an accuracy of 91% and an f-measure of 84% (Table 5.7). Table 5.6 indicates that the artificial neural network model correctly predicted 13 of 15 negative outcomes. These results

indicate that at this stage of the year, the model is no longer assuming a positive prediction. Therefore, an effective prediction can be made at this stage.

6.2.3.2 Decision Trees

A decision tree model was created for each of the three datasets (Section 4.4). Decision trees achieved an accuracy of 77% and an f-measure of 33% for the Quarter 1 dataset (Table 5.9). Table 5.8 indicates that the decision tree model correctly predicted only 3 of 10 negative outcomes. These results indicate that at this stage of the year, the model is mostly assuming a positive prediction. This is due to the imbalanced nature of the dataset. Therefore, an effective prediction cannot be made at this stage.

When trained on the Quarter 2 dataset, decision trees yielded a significantly higher performance with an accuracy of 92% and an f-measure of 88% (Table 5.11). Table 5.10 indicates that the decision tree model correctly predicted 14 of 15 negative outcomes. These results indicate that at this stage of the year, the model is no longer assuming a positive prediction. In contrast to the artificial neural network model, the decision tree model is able to predict negative outcomes with a 93% accuracy. Therefore, an effective prediction can definitely be made at this stage.

When trained on the Quarter 3 dataset, decision trees showed a slight increase when compared to the previous dataset. The model achieved an accuracy of 96% and an f-measure of 93% (Table 5.13). Table 5.12 indicates that the decision tree model correctly predicted 13 of 14 negative outcomes. Similar to when trained on the

previous dataset, this model is able to predict both positive and negative outcomes accurately. Therefore, an effective prediction can be made at this stage.

6.2.3.3 Naïve Bayes Classifier

A naïve Bayes classification model was created for each of the three datasets (Section 4.4). Naïve Bayes classifiers achieved an accuracy of 74% and an f-measure of 42% for the Quarter 1 dataset (Table 5.15). Table 5.14 indicates that the naïve Bayes model correctly predicted only 5 of 12 negative outcomes. These results indicate that at this stage of the year, the naïve Bayes model is mostly assuming a positive prediction. This is due to the imbalanced nature of the dataset. Therefore, an effective prediction cannot be made at this stage.

When trained on the Quarter 2 dataset, naïve Bayes classifiers yielded a higher performance with an accuracy of 83% and an f-measure of 67% (Table 5.17). Table 5.16 indicates that the naïve Bayes model correctly predicted 9 of 15 negative outcomes. These results indicate that at this stage of the year, the model is still mostly assuming a positive prediction, although, to a lesser degree. Therefore, a somewhat effective prediction can be made at this stage.

When trained on the Quarter 3 dataset, naïve Bayes classifiers showed an increase in both accuracy and f-measure when compared to the previous two datasets. The naïve Bayes model achieved an accuracy of 94% and an f-measure of 86% (Table 5.19). Table 5.18 indicates that the model correctly predicted 9 of 10 negative outcomes. These results indicate that at this stage of the year, the naïve Bayes model

is no longer assuming a positive prediction. Therefore, an effective prediction can be made at this stage.

Having discussed the performance of each prediction model separately, one can now compare them based on each dataset. Figure 5.2 plots the comparative f-measures for each dataset.

6.2.3.4 Comparative Discussion

From this study it is evident that using more than one modelling technique allows for comparison of performances. There are many factors that influence the performance of different models. Domingos (2012) states that you should learn many models and not just one. This enabled the study to find results, which it would not have discovered otherwise. This leads to the third guideline relating to data mining and evaluation (Guideline 3.3):

If possible, implement and compare different machine learning techniques to produce relevant prediction models and choose one based on the desired level of detail.

Artificial neural networks achieved the best performance when trained on the Quarter 1 dataset. However, as with the other techniques, an effective prediction cannot be made this early in the year.

When trained on the Quarter 2 dataset, artificial neural networks and naïve Bayes classifiers fall short and are unable to make an effective prediction at this stage of the year. However, artificial neural networks and naïve Bayes classifiers are able to

produce a probability of a prediction, as mentioned in Section 3.5 and Section 3.7. In contrast, decision trees are only able to produce a binary prediction. This leads to the first sub-guideline of Guideline 3.3 (Guideline 3.3.1):

If the probability of a prediction is required, make use of artificial neural networks and naïve Bayes classifiers.

Decision trees, however, achieved a significantly higher performance compared to the other techniques when trained on the Quarter 2 dataset. From the results it is evident that an effective prediction can be made at this stage of the year when using decision trees. By this stage, decision trees achieved an accuracy higher than both the other prediction models in the third quarter. Therefore, decision trees should be used to predict student group academic performance. This leads to the second sub-guideline of Guideline 3.3 (Guideline 3.3.2):

If prediction accuracy is important, decision trees should be considered when predicting student group academic performance.

As with the previous dataset, decision trees achieved the best performance when trained on the Quarter 3 dataset. However, all machine learning techniques are able to make an effective prediction at this point.

Figure 5.3 plots the comparative variances for the models when trained on each of the datasets. The variance indicates how likely the model is to achieve the same performance on new data. On average, the variances for each of the models is relatively high. This is as a result of the size and imbalanced distribution of the

dataset and indicates overfitting. However, decision trees achieved the lowest variances and fall within an acceptable range.

6.3 Summary of Guidelines

The guidelines discussed in this chapter are summarised in Figure 6.3:

GUIDELINES	
1. Raw Data and Selection	
1.1	Modify the existing data storage methods to allow for the collection of additional attributes when appropriate.
1.2	Balance the input dataset through the use of appropriate techniques to prevent an unbalanced target distribution.
2. Pre-Processing and Transformation	
2.1	Automate the preparation and pre-processing of the data by making use of an application.
2.2	Ensure that all data stores contain a corresponding unique identifier.
2.3	Store assessment mark file data in a consistent predetermined format.
3. Data Mining and Evaluation	
3.1	Evaluate the prediction models based on the f-measure as the final evaluation criterion.
3.2	Cross validate using k-fold cross validation when testing the prediction models and record the variance of the models.
3.3	If possible, implement and compare different machine learning techniques to produce relevant prediction models and choose one based on the desired level of detail.
3.3.1	If the probability of a prediction is required, make use of artificial neural networks and Naïve Bayes classifiers.
3.3.2	If prediction accuracy is important, decision trees should be considered when predicting student group academic performance.

Figure 6.3: Summary of Guidelines

By following these guidelines, one is able to predict student group academic performance successfully. If the purpose of following these guidelines is to

implement an intervention for student groups at risk, then various considerations need to be considered.

One main consideration is to choose an appropriate time during the year to build the prediction models. When making this decision it is important to consider the balance of available data with the timing for intervention. In this study, by the third iteration, student group academic performance was predicted with a high degree of accuracy. However, this might be too late for an effective intervention to be made. In contrast, the first iteration would be the best stage for intervention as there is more time for the intervention to make a difference. However, as seen in this study, in the first iteration there is not enough data to predict student group academic performance effectively.

Therefore, it is evident that there is a trade-off between the data available and the time remaining in which to make an intervention. In this study a highly accurate prediction was able to be made using data from the second iteration and before. This provides a good balance between available data and the time remaining in which to make an intervention. This study therefore shows that by following these guidelines it is possible to make an accurate prediction of student group academic performance with sufficient time left in the year for intervention with project groups at risk.

6.4 Conclusion

This chapter followed the steps of extracting knowledge from data used in education data mining and learning analytics to discuss the literature and findings

of this study. The selection of the raw data was first discussed, followed by a discussion surrounding data pre-processing and transformation. Finally, each of the machine learning techniques was discussed individually and comparatively with regard to the analysis performed in chapter 5.

Each of these discussions resulted in guidelines, which aimed to address problems and make recommendations identified within. These guidelines were then summarised and discussed. The creation of these guidelines achieves the primary objective of this study which was:

To provide guidelines for the use of machine learning techniques to predict academic performance of student project groups.

These guidelines are intended to be used in future studies and experiments on using machine learning to predict student academic performance both within and outside the context of the Nelson Mandela University. The guidelines provided serve as a baseline and should be altered and expanded to fit the context of each case of use.

This study is concluded in Chapter 7 by summarising the chapters, describing how each of the objectives was met, summarising the contribution of this study and finally, providing suggestions for future research.

Chapter 7: Conclusion

7.1 Introduction

The previous chapter followed the steps of extracting knowledge from data to discuss the literature and findings of this study. From these discussions, guidelines were provided in order to address the primary objective of this study. This study investigated the lack of data-driven decision-making for predicting student group academic performance at the Nelson Mandela University. The main aim of this study was to investigate how educational data could be used for the early identification of project groups that might be at risk of failure.

This chapter provides a summary of the previous chapters (Section 7.2). The research objectives are reviewed in order to determine whether the study was successful (Section 7.3). The theoretical and practical contributions of the study are emphasised (Section 7.4). Finally, recommendations are presented on how this work can be applied and expanded to other contexts, and how it may be applicable to future research (Section 7.5).

7.2 Summary of Chapters

A literature review was undertaken in Chapter 2, which addressed secondary objectives SO1: *To identify common attributes of student activities used to predict academic performance,* and SO2: *To identify what data and attributes are currently available in Moodle to be used for predicting student group academic performance.*

This literature review discussed four main topics, namely learning management systems, learning analytics, education data mining and performance predicting attributes in existing research. Learning management systems were defined and discussed. Various uses and benefits of LMS were mentioned in this chapter with specific focus on Moodle as this is the learning management system used by the Nelson Mandela University where this study took place. The data that is stored in these Moodle activity logs was identified together with the format in which it is stored. Learning analytics and educational data mining were both defined in this chapter. The processes and affected stakeholders were also discussed together with some of the challenging issues. Finally, a number of performances predicting attributes were summarised. This summary was formed from existing research in the field of predicting student academic performance.

A literature further review was conducted in Chapter 3, which addressed secondary objective SO3: *To determine the machine learning techniques commonly used by educational data mining and learning analytics and argue their relevance to a specific data set.* This literature discussed the main topics associated with machine learning: data pre-processing, an introduction to machine learning, techniques used in previous studies, a discussion on each of the selected machine learning techniques and model evaluation methods. Data pre-processing was defined, and the process discussed based on the key steps, namely data cleaning, data integration, data selection and data transformation. Machine learning as a concept was introduced by discussing the applications and different variations. Machine learning techniques used in predicting student academic models were summarised. This

summary was formed from existing research in the field. It revealed that decision trees, neural networks and naïve Bayes classifiers were the best performing methods. Each of these methods was then defined and discussed in detail. Finally, methods were identified for evaluating prediction models.

Chapter 4 described the experiment design and prediction model creation process, which addressed secondary objective S04: *To use the selected machine learning techniques to create prediction models to predict student group academic performance.* This chapter described, in detail, each step of the data pre-processing performed by this study. The code and output data for each step of the process described in Chapter 3 was explained. The models were then designed and trained on the pre-processed data.

The results of the prediction models were presented and analysed in Chapter 5. The aim of this chapter was to provide a baseline for each of the prediction models to be discussed, both individually and comparatively, in Chapter 6. Therefore, secondary objective SO5 was addressed: *To evaluate and compare the performance of selected machine learning techniques in predicting student group academic performance.*

7.3 Accomplishment of Research Objectives

The research objectives of this study were outlined in Section 1.3 as follows:

7.3.1 Primary Objective

To provide guidelines for the use of machine learning techniques to predict academic performance of student project groups.

The secondary objectives provided the necessary literature and findings from the experiment conducted to address the primary objective of this research. Chapter 6 provides a final discussion of the modelling process undertaken by this study. Through this discussion, guidelines were developed to address the use of machine learning techniques to predict the academic performance of student project groups.

7.3.2 Secondary Objective

SO_1: *To identify common attributes of student activities used to predict academic performance.*

This objective was addressed in Chapter 2 where a literature review was performed, which identified commonly used attributes in existing research on predicting student academic performance.

SO_2: *To identify what data and attributes are currently available in Moodle to be used for predicting student group academic performance.*

This objective was addressed in Chapter 2 where an investigation was conducted with the goal of identifying the data contained in Moodle log files, which could be used by the selected machine learning techniques.

SO_3: *To determine the machine learning techniques commonly used by educational data mining and learning analytics and to argue their relevance to a specific data set.*

This objective was achieved in Chapter 3 where a literature review was conducted on existing research, which identified machine learning techniques that performed well in a similar context and on a similar dataset. Each of the identified techniques was then discussed in detail, expanded on in a literature review.

SO_4: *To use the selected machine learning techniques to create prediction models to predict student group academic performance.*

This objective was achieved in Chapter 4 where predictive models were created based on the machine learning techniques identified in SO_3. These models were trained using the data identified in SO_1 and SO_2.

SO_5**:** *To evaluate and compare the performance of selected machine learning techniques in predicting student group academic performance.*

This objective was achieved in Chapter 5 and Chapter 6 where the resulting performance metrics of the prediction models were listed, analysed and discussed individually and comparatively.

7.4 Summary of Contribution

This study aimed to create guidelines for the use of machine learning techniques to predict academic performance of student project groups at the Nelson Mandela University. The scope of this study was to use the data contained in Moodle action log files and assessment mark files provided by the lecturer. The data provided was for the third-year capstone project module (ONT3660) for 2016 and 2017.

The collection and pre-processing of this data was discussed in Chapter 4 with the attribute selection made based on the available data and a literature review performed in Chapter 2. This literature review identified common attributes used to predict student academic performance. The information gained from this review helped in building a rich dataset as well as in constructing the guidelines.

The pre-processed data was used to create multiple prediction models based on three different machine learning techniques, namely artificial neural networks, decision trees and naïve Bayes classifiers. These machine learning techniques were identified in Chapter 3 where a literature review was conducted on previous work. Three models were created for each of these machine learning techniques in order to identify how early in the year student group academic performance could be predicted. It was found that by using the available data, an accurate prediction could be made by the second of the four iterations that span the year.

In summary, the guidelines developed cover the three key aspects of learning analytics and educational data mining, namely raw data and selection, pre-

processing and transformation, and data mining and evaluation. In conclusion, this study showed that by following the guidelines proposed in Chapter 6, an effective prediction of student group academic performance could be performed. These early predictions allow for the early identification of student project groups at risk. With these groups identified, applicable interventions could be implemented as early as the second iteration, which is halfway through the year.

7.5 Limitations and Suggestions for Further Research

This study used a number of machine learning techniques to create various prediction models. There are possibly other machine learning techniques that could perform better. In further research these additional machine learning techniques could be considered.

Although this study was limited by the data used in the modelling process, an effective prediction could not be made. The literature within this study suggested that a richer dataset would produce better results. It is therefore recommended that in future research additional data should be considered, as stipulated by Guideline 1.1. This will result in more effective prediction models with the possibility of a prediction being made earlier in the year.

References

Adam, S., & Nel, D. (2009). Blended and online learning: student perceptions and performance. *Interactive technology and smart education*.

Abu Tair, M. M., & El-Halees, A. M. (2012). Mining educational data to improve students' performance: a case study. *Mining educational data to improve students' performance: a case study*, *2*(2).

Ahmed, A. B. E. D., & Elaraby, I. S. (2014). Data mining: A prediction for student's performance using classification method. *World Journal of Computer Application and Technology*, *2*(2), 43-47.

Al-Twijri, M. I., & Noaman, A. Y. (2015). A new data mining model adopted for higher institutions. *Procedia Computer Science*, *65*, 836-844.

Alpaydin, E. Introduction to machine learning. 2004. 1-327.

Baker, B. M. (2007). *A conceptual framework for making knowledge actionable through capital formation* (Doctoral dissertation, University of Maryland University College).

Baradwaj, B. K., & Pal, S. (2012). Mining educational data to analyze students' performance. *International Journal of Advanced Computer Science and Applications,* 2(16), 63-69

Beer, C., Clark, K., & Jones, D. (2010). Indicators of engagement. In *Curriculum, technology & transformation for an unknown future. Proceedings from Ascilite Sydney*, 75-86.

Bersin, J. (2004). *The blended learning book: Best practices, proven methodologies, and lessons learned*. John Wiley & Sons.

Bishop, M. (2006) *Pattern Recognition and Machine Learning*. Cambridge: Springer Science.

Blatchford, P., Bassett, P., & Brown, P. (2011). Examining the effect of class size on classroom engagement and teacher–pupil interaction: Differences in relation to pupil prior attainment and primary vs. secondary schools. *Learning and Instruction*, *21*(6), 715-730.

Boughey, C. (2003). From equity to efficiency: Access to higher education in South Africa. *Arts and Humanities in Higher Education*, *2*(1), 65-71.

Campbell, J. P., DeBlois, P. B., & Oblinger, D. G. (2007). Academic analytics: A new tool for a new era. *EDUCAUSE review*, *42*(4), 40.

Cavus, N., Uzunboylu, H., & Ibrahim, D. (2007). Assessing the success rate of students using a learning management system together with a collaborative tool in web-based teaching of programming languages. *Journal of educational computing research*, *36*(3), 301-321.

Cheewaprakobkit, P. (2015). Predicting student academic achievement by using the decision tree and neural network techniques. *Catalyst*, 12(2), 34-43.

Dahlstrom, E., & Bichsel, J. (2014). ECAR Study of Undergraduate Students and Information Technology, 2014. *Educause*.

Daud, A., Aljohani, N. R., Abbasi, R. A., Lytras, M. D., Abbas, F., & Alowibdi, J. S. (2017, April). Predicting student performance using advanced learning analytics. In *Proceedings of the 26th international conference on world wide web companion* (pp. 415-421). International World Wide Web Conferences Steering Committee.

Dietz-Uhler, B., & Hurn, J. E. (2013). Using learning analytics to predict (and improve) student success: A faculty perspective. *Journal of interactive online learning*, *12*(1), 17-26.

Do, T. (2007). Towards Simple, Easy To Understand, An Interactive Decision Tree Algorithm. *College Inf. Technol*., 6(1).

Domingos, P. M. (2012). A few useful things to know about machine learning. *Communications of the acm*, *55*(10), 78-87.

Ebardo, R. A., & Valderama, A. M. C. (2009, December). The effect of web-based learning management system on knowledge acquisition of information technology students at Jose Rizal University. In *Proceeding of 6th International Conference on E-learning for Knowledge-based Society,* Bangkok, Thailand.

Elias, T. (2011). Learning Analytics: Definitions, Processes and Potential. *Learning*, 23, 134–148.

Estacio, R. R., & Raga Jr, R. C. (2017). Analyzing students' online learning behavior in blended courses using Moodle. *Asian Association of Open Universities Journal*, *12*(1), 52-68.

Ferguson, R. (2012). Learning analytics: drivers, developments and challenges. *International Journal of Technology Enhanced Learning*, *4*(5/6), 304-317.

Freeman, J. A., & Skapura, D. M. (1991). *Neural networks: algorithms, applications, and programming techniques*. Addison Wesley Longman Publishing Co., Inc.

Fry, K. (2001). E-learning markets and providers: some issues and prospects. *Education+ Training*, *43*(4/5), 233-239.

Glorot, X., Bordes, A., & Bengio, Y. (2011, June). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 315-323).

Greller, W., & Drachsler, H. (2012). Translating learning into numbers: A generic framework for learning analytics. *Educational Technology & Society*, 15 (3), 42–57.

Grubbs, F. E. (1969). Procedures for detecting outlying observations in samples. *Technometrics*, *11*(1), 1-21.

Hamsa, H., Indiradevi, S., & Kizhakkethottam, J. J. (2016). Student academic performance prediction model using decision tree and fuzzy genetic algorithm. *Procedia Technology*, *25*, 326-332.

Han, J., Kamber, M. & Pei, J. (2011). *Data mining: concepts and techniques*. Elsevier.

Herzog, S. (2006). Estimating student retention and degree-completion time: Decision trees and neural networks vis-à-vis regression. *New Directions for Institutional Research*, 2006(131), 17-33.

Hijazi, S. T., & Naqvi, S. M. M. (2006). Factors Affecting Students' Performance. *Bangladesh e-journal of Sociology*, *3*(1).

Ibrahim, Z. and Rusli, D. (2007). Predicting students' academic performance: comparing artificial neural network, decision tree and linear regression. 21st Annual SAS Malaysia Forum, 5th September 2007, Kuala Lumpur.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Kane, T., Kerr, K., & Pianta, R. (2014). *Designing teacher evaluation systems: New guidance from the measures of effective teaching project*. John Wiley & Sons.

Kaur, P., Singh, M., & Josan, G. S. (2015). Classification and prediction-based data mining algorithms to predict slow learners in education sector. *Procedia Computer Science*, *57*, 500-508.

KDnuggets. (2019). 7 Techniques to Handle Imbalanced Data. Retrieved from https://www.kdnuggets.com/2017/06/7-techniques-handle-imbalanced-data.html.

Khoonsari, P. E., & Motie, A. (2012). A comparison of efficiency and robustness of ID3 and C4. 5 algorithms using dynamic test and training data sets. *International Journal of Machine Learning and Computing*, *2*(5), 540.

Klösgen, W., & Zytkow, J. M. (2002). *Handbook of data mining and knowledge discovery*. Oxford University Press, Inc.

Kotsiantis, S., & Kanellopoulos, D. (2006). Association rules mining: A recent overview. *GESTS International Transactions on Computer Science and Engineering*, *32*(1), 71-82.

Kotsiantis, S., Pierrakeas, C., & Pintelas, P. (2004). Predicting Students' Performance. In Distance Learning Using Machine Learning Techniques. *Applied Artificial Intelligence*, *18*(5), 411-426.

Lee, M. L., Hsu, W., & Kothari, V. (2004). Cleaning the spurious links in data. *IEEE Intelligent Systems*, *19*(2), 28-33.

Liu, D. Y. T., Froissard, J. C., Richards, D., & Atif, A. (2015). An enhanced learning analytics plugin for Moodle: student engagement and personalised intervention.

Lopes, A. P. (2011). Teaching with Moodle in higher education. *INTED 2011*.

Macfadyen, L. P., & Dawson, S. (2010). Mining LMS data to develop an "early warning system" for educators: A proof of concept. *Computers & education*, *54*(2), 588-599.

May, T. A. (2011). *Analytics, University 3.0, and the future of information technology*. EDUCAUSE.

Mitchell, T. M. (1997). Does machine learning really work? *AI magazine*, *18*(3), 11-11.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). Foundations of Machine Learning. Adaptive computation and machine learning. *MIT Press*, *31*, 32.

Monarch Media Incorporated. (2010). OpenSource Learning Management Systems: Sakai and Moodle Monarch. Santa Cruz: Business White Paper.

Moodle. (2017). Moodledocs. Retrieved from https://docs.moodle.org.

Munguatosha, G.M., Muyinda, P.B., & Lubega, T. J. (2011). A social networked learning adoption model for higher education institutions in developing countries. *On the Horizon*, *19*(4), 307-320.

Nair, S. C., & Patil, R. (2012). A study on the impact of learning management systems on students of a university college in Sultanate of Oman. *International Journal of Computer Science Issues (IJCSI)*, *9*(2), 379.

Nghe, N. T., Janecek, P., & Haddawy, P. (2007, October). A comparative analysis of techniques for predicting academic performance. In *2007 37th annual frontiers*

in education conference-global engineering: knowledge without borders, opportunities without passports (pp. T2G-7). IEEE.

Nyce, C. (2007). Predictive analytics white paper. *American Institute for CPCU. Insurance Institute of America*, 9-10.

Olivier, M. S. (2009). Information technology research: A practical guide for computer science and informatics. *Van Schaik*.

Olmos, M. M., & Corrin, L. (2012). *Learning analytics: A case study of the process of design of visualizations.*

Osmanbegović, E., & Suljić, M. (2012). Data mining approach for predicting student performance. *Economic Review: Journal of Economics and Business*, *10*(1), 3-12.

Pappas, C. (2017). What is a learning management system? LMS basic functions and features you must know. *eLearning Industry*.

Patil, N., Lathi, R., & Chitre, V. (2012). Comparison of C5. 0 & CART classification algorithms using pruning technique. *Int. J. Eng. Res. Technol*, *1*(4), 1-5.

Powers, D. M. (2011). *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.*

Quinlan, J. R. (1990). Probabilistic decision trees. In *Machine Learning* (pp. 140-152). Morgan Kaufmann.

Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, *23*(4), 3-13.

Ramaswami, M., & Bhaskaran, R. (2010). *A CHAID based performance prediction model in educational data mining.*

Ramesh, V., Parkavi, P., & Ramar, K. (2013). Predicting student performance: a statistical and data mining approach. *International journal of computer applications*, *63*(8), 35-39.

Rokach, L., & Maimon, O. (2005). Top-down induction of decision trees classifiers-a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *35*(4), 476-487.

Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert systems with applications*, *33*(1), 135-146.

Romero, C., Ventura, S., & García, E. (2008). Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, *51*(1), 368-384.

Sakai. (2018). Sakai Learning Management System. Retrieved from https://www.sakailms.org.

SAS. (2017). Predictive Analytics: What it is and why it matters, SAS. Retrieved from https://www.sas.com/en_us/insights/analytics/predictive-analytics.html.

Shah, N. S. (2012). Predicting Factors That Affect Students' Academic Performance by Using Data Mining Techniques. *Pakistan Business Review*, 13(4), 631-638.

Shahiri, A. M., Husain, W., & Rashid, N. A. (2015). A review on predicting student's performance using data mining techniques, *Procedia Computer Science*.

Shalem, B., Bachrach, Y., Guiver, J., & Bishop, C. M. (2014, September). Students, teachers, exams and MOOCs: Predicting and optimizing attainment in web-based education using a probabilistic graphical model. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 82-97). Springer, Berlin, Heidelberg.

Shmueli, G., & Koppius, O. R. (2011). Predictive analytics in information systems research. *MIS quarterly*, 553-572.

Siemens, G., & Long, P. (2011). Penetrating the fog: Analytics in learning and education. *EDUCAUSE review*, *46*(5), 30.

Simeunović, V. and Preradović, L. (2014). Using data mining to predict success in studying. *Croatian Journal of Education*, 16(2), 491-523.

SkyPrep. (2017). SkyPrep: Online Training Software. Retrieved from https://www.skyprep.com.

Smith, V. C., Lange, A., & Huston, D. R. (2012). Predictive modeling to forecast student outcomes and drive effective interventions in online community college courses. *Journal of Asynchronous Learning Networks*, *16*(3), 51-61.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929-1958.

Sugiyama, M. (2015). *Introduction to statistical machine learning*. Morgan Kaufmann.

Tang, Y., & Salakhutdinov, R. R. (2013). Learning stochastic feedforward neural networks. In *Advances in Neural Information Processing Systems* (pp. 530-538).

Uma, K., & Hanumanthappa, M. (2017). Data Collection Methods and Data Pre-processing Techniques for Healthcare Data Using Data Mining. *International Journal of Scientific & Engineering Research*, *8*(6), 1131-1136.

Unwin, T., Kleessen, B., Hollow, D., Williams, J., Oloo, L. M., Alwala, J., Mutimucuio, I. Eduardo, F. & Muianga, X. (2010). Digital Learning Management Systems in Africa: Myths and Realities. *Open Learning: The Journal of Open and Distance Learning*, 25(1), 5-23.

Van Hulse, J., & Khoshgoftaar, T. (2009). Knowledge discovery from imbalanced and noisy data. *Data & Knowledge Engineering*, *68*(12), 1513-1542.

Van Harmelen, M., & Workman, D. (2012). Analytics for learning and teaching. *CETIS Analytics Series*, *1*(3), 1-40.

Van Niekerk, J., & Webb, P. (2016). The effectiveness of brain-compatible blended learning material in the teaching of programming logic. *Computers & Education*, *103*, 16-27.

Von Alan, R. H., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, *28*(1), 75-105.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Yadav, S. K., Bharadwaj, B., & Pal, S. (2012). Mining Education data to predict student's retention: a comparative study. *International Journal of Computer Science and Information Security*, 10(1), 113-117.

Yu, T., & Jo, I. H. (2014). Educational technology approach toward learning analytics: Relationship between student online behavior and learning performance in higher education. In *Proceedings of the Fourth International Conference on Learning Analytics and Knowledge* (pp. 269-270). ACM.

Zekić-Sušac, M., Frajman-Jakšić, A. & Drvenkar, N. (2009). Neuron networks and trees of decision-making for prediction of efficiency in studies, *Ekonomski vjesnik (Econviews*), 22(2), 314-327.

Appendix A: Raw Data

Moodle Action Logs

Date	Time	User Name	Affected User	Event Context	Component	Event Name	Description	Origin	IP Address
2016/05/05	12:11		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '21689' viewed the course with id '2579'.	web	
2016/05/05	11:46		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '29032' viewed the 'resource' activity with course module id '93091'.	web	
2016/05/05	11:46		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '29032' viewed the course with id '2579'.	web	
2016/05/05	11:39		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '15020' viewed the course with id '2579'.	web	
2016/05/05	11:35		-	File: Minutes of Meetings Template	File	Course module viewed	The user with id '21763' viewed the 'resource' activity with course module id '93048'.	web	
2016/05/05	11:34		-	File: Story Board Assessment Rubric	File	Course module viewed	The user with id '21763' viewed the 'resource' activity with course module id '103677'.	web	
2016/05/05	11:32		-	Assignment: UPLOAD Storyboard Designs	Assignment	The status of the submission has been viewed.	The user with id '30247' has viewed the submission status page for the assignment with course module id '93092'.	web	
2016/05/05	11:32		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '30247' viewed the 'resource' activity with course module id '93091'.	web	
2016/05/05	11:32		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '30247' viewed the 'resource' activity with course module id '93090'.	web	
2016/05/05	11:31		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '30247' viewed the course with id '2579'.	web	
2016/05/05	11:28		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '13612' viewed the course with id '2579'.	web	
2016/05/05	11:21		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '12944' viewed the course with id '2579'.	web	
2016/05/05	11:21		-	Assignment: Upload Sample Reports	Assignment	The status of the submission has been viewed.	The user with id '12944' has viewed the submission status page for the assignment with course module id '93100'.	web	
2016/05/05	11:21		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '12944' viewed the course with id '2579'.	web	
2016/05/05	11:17		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '4138' viewed the 'resource' activity with course module id '93091'.	web	
2016/05/05	11:17		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '4138' viewed the 'resource' activity with course module id '93090'.	web	
2016/05/05	11:17		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '4138' viewed the course with id '2579'.	web	
2016/05/05	11:16		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '13009' viewed the 'resource' activity with course module id '93090'.	web	
2016/05/05	11:16		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '13009' viewed the course with id '2579'.	web	
2016/05/05	11:12		-	File: Story Board Assessment Rubric	File	Course module viewed	The user with id '21763' viewed the 'resource' activity with course module id '103677'.	web	
2016/05/05	11:01		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '7133' viewed the 'resource' activity with course module id '93090'.	web	
2016/05/05	11:01		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '7133' viewed the 'resource' activity with course module id '93091'.	web	
2016/05/05	11:01		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '7133' viewed the course with id '2579'.	web	
2016/05/05	11:01		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '21533' viewed the 'resource' activity with course module id '93091'.	web	
2016/05/05	11:01		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '21533' viewed the course with id '2579'.	web	
2016/05/05	10:59		-	Assignment: UPLOAD Storyboard Designs	Assignment	The status of the submission has been viewed.	The user with id '6953' has viewed the submission status page for the assignment with course module id '93092'.	web	
2016/05/05	10:59		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '6953' viewed the course with id '2579'.	web	
2016/05/05	10:55		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '21763' viewed the 'resource' activity with course module id '93091'.	web	
2016/05/05	10:55		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '21763' viewed the 'resource' activity with course module id '93090'.	web	
2016/05/05	10:55		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '21763' viewed the course with id '2579'.	web	
2016/05/05	10:55		-	File: Minutes of Meetings Template	File	Course module viewed	The user with id '30104' viewed the 'resource' activity with course module id '93048'.	web	
2016/05/05	10:55		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '30104' viewed the course with id '2579'.	web	
2016/05/05	10:53		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '29097' viewed the 'resource' activity with course module id '93091'.	web	
2016/05/05	10:51		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '29097' viewed the 'resource' activity with course module id '93090'.	web	
2016/05/05	10:51		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '29097' viewed the course with id '2579'.	web	
2016/05/05	10:51		-	Assignment: UPLOAD Storyboard Designs	Assignment	The status of the submission has been viewed.	The user with id '29097' has viewed the submission status page for the assignment with course module id '93092'.	web	
2016/05/05	10:50		-	File: Story Board Assessment Rubric	File	Course module viewed	The user with id '29097' viewed the 'resource' activity with course module id '103677'.	web	
2016/05/05	10:50		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '29097' viewed the 'resource' activity with course module id '93090'.	web	
2016/05/05	10:50		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '29097' viewed the course with id '2579'.	web	
2016/05/05	10:45		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '21689' viewed the 'resource' activity with course module id '93091'.	web	
2016/05/05	10:45		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '21689' viewed the 'resource' activity with course module id '93090'.	web	
2016/05/05	10:45		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '21689' viewed the course with id '2579'.	web	
2016/05/05	10:29		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '29128' viewed the 'resource' activity with course module id '93090'.	web	
2016/05/05	10:29		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '29128' viewed the course with id '2579'.	web	
2016/05/05	10:29		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '13612' viewed the course with id '2579'.	web	
2016/05/05	10:29		-	Assignment: Upload Sample Reports	Assignment	The status of the submission has been viewed.	The user with id '13612' has viewed the submission status page for the assignment with course module id '93100'.	web	
2016/05/05	10:27		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '21763' viewed the course with id '2579'.	web	
2016/05/05	10:27		-	Assignment: Upload Sample Reports	Assignment	The status of the submission has been viewed.	The user with id '21763' has viewed the submission status page for the assignment with course module id '93100'.	web	
2016/05/05	10:26		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '21524' viewed the 'resource' activity with course module id '93091'.	web	
2016/05/05	10:26		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '29128' viewed the course with id '2579'.	web	
2016/05/05	10:26		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '21763' viewed the course with id '2579'.	web	
2016/05/05	10:22		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '21533' viewed the 'resource' activity with course module id '93090'.	web	
2016/05/05	10:22		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '21533' viewed the course with id '2579'.	web	
2016/05/05	10:19		-	File: Story Board Assessment Rubric	File	Course module viewed	The user with id '21689' viewed the 'resource' activity with course module id '103677'.	web	
2016/05/05	10:18		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '21689' viewed the course with id '2579'.	web	
2016/05/05	10:17		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '13612' viewed the course with id '2579'.	web	
2016/05/05	10:16		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '29175' viewed the 'resource' activity with course module id '93090'.	web	
2016/05/05	10:16		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '29175' viewed the 'resource' activity with course module id '93091'.	web	
2016/05/05	10:16		-	File: Story Board Assessment Rubric	File	Course module viewed	The user with id '29175' viewed the 'resource' activity with course module id '103677'.	web	
2016/05/05	10:16		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '29175' viewed the course with id '2579'.	web	
2016/05/05	10:14		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '13182' viewed the course with id '2579'.	web	
2016/05/05	10:13		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '13612' viewed the course with id '2579'.	web	
2016/05/05	10:07		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '15020' viewed the 'resource' activity with course module id '93091'.	web	
2016/05/05	10:06		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '15020' viewed the 'resource' activity with course module id '93090'.	web	
2016/05/05	10:03		-	File: WEEK 11 Slides - StoryBoard Designs	File	Course module viewed	The user with id '15020' viewed the 'resource' activity with course module id '103676'.	web	
2016/05/05	10:03		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '29077' viewed the course with id '2579'.	web	
2016/05/05	10:02		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '15020' viewed the course with id '2579'.	web	
2016/05/05	10:01		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '20520' viewed the 'resource' activity with course module id '93090'.	web	
2016/05/05	10:01		-	File: Sample Storyboard Design	File	Course module viewed	The user with id '20520' viewed the 'resource' activity with course module id '93091'.	web	
2016/05/05	10:00		-	Module: ONT3660: 3rd year Software Development Project 2016	System	Course viewed	The user with id '20520' viewed the course with id '2579'.	web	

Assessment Mark Files

Student Number	Student Surname	First Name	Group Name	Project Supervisor	Project Idea	IT1 TOTAL (10%)	IT2 TOTAL (30%)	IT3 TOTAL (30%)	IT4 TOTAL (30%)	FINAL MARK
214040119			Incognitus	Dieter Steenberg	Addict Online Support System	78,6	83,3	94,3	100	91,14
215068505			Incognitus	Dieter Steenberg	Addict Online Support System	78,6	83,3	94,3	100	91,14
215093089			Incognitus	Dieter Steenberg	Addict Online Support System	78,6	83,3	94,3	100	91,14
215096223			Incognitus	Dieter Steenberg	Addict Online Support System	78,6	83,3	94,3	100	91,14
212244574			Sandstorm	Cheryl Schroder	Number Crumble Educational System	81,2	86,8	93,6	95,8	90,98
215131657			Sandstorm	Cheryl Schroder	Number Crumble Educational System	81,2	86,8	93,6	95,8	90,98
213300893			TriSoft	Lynn Futcher	Property Deal Tracking System	79,2	77,6	92,6	93	86,88
214157679			TriSoft	Lynn Futcher	Property Deal Tracking System	79,2	77,6	92,6	93	86,88
214212483			TriSoft	Lynn Futcher	Property Deal Tracking System	79,2	77,6	92,6	93	86,88
215046862			KeyDevelop	Yiota Moutzouris	School Management System	77,4	80	90,6	91,2	86,28
215062469			KeyDevelop	Yiota Moutzouris	School Management System	77,4	80	90,6	91,2	86,28
215044215			Unity	Lynn Futcher	Educational Community Programs (IGEMS)	83	78,8	87,3	87,6	84,41
215149866			Unity	Lynn Futcher	Educational Community Programs (IGEMS)	83	78,8	87,3	87,6	84,41
215161033			Unity	Lynn Futcher	Educational Community Programs (IGEMS)	83	78,8	87,3	87,6	84,41
215362861			Unity	Lynn Futcher	Educational Community Programs (IGEMS)	83	78,8	87,3	87,6	84,41
215030281			Genesys	Johan	Eminence	75,4	74,6	86,4	89	82,54
215033000			Genesys	Johan	Eminence	75,4	74,6	86,4	89	82,54
215144848			Genesys	Johan	Eminence	75,4	74,6	86,4	89	82,54
212242822			Pixel Pushers	Johan	Eidolon	81	75,6	82,5	80,8	79,77
215157907			Pixel Pushers	Johan	Eidolon	81	75,6	82,5	80,8	79,77
215074068			SEAM-Tech1	Lynn Futcher	Student Counselling System	75	71,4	83,2	84	79,08
215121872			SEAM-Tech1	Lynn Futcher	Student Counselling System	75	71,4	83,2	84	79,08
215285085			SEAM-Tech1	Lynn Futcher	Student Counselling System	75	71,4	83,2	84	79,08
214154874			Bits and Bytes	Cinga	Social Campus Football League	68,8	69,9	79,3	87,2	77,8
214158128			Bits and Bytes	Cinga	Social Campus Football League	68,8	69,9	79,3	87,2	77,8
214293009			Bits and Bytes	Cinga	Social Campus Football League	68,8	69,9	79,3	87,2	77,8
214313239			Bits and Bytes	Cinga	Social Campus Football League	68,8	69,9	79,3	87,2	77,8
211056308			PharmaTech	Cheryl Schroder	Pharmacy System	73	72,8	75	86,6	77,62
212273582			PharmaTech	Cheryl Schroder	Pharmacy System	73	72,8	75	86,6	77,62
214167089			Archonix Games	Johan	Battle for Archonia	70,4	82,8	73,6	78,4	77,48
215125304			Archonix Games	Johan	Battle for Archonia	70,4	82,8	73,6	78,4	77,48
215134982			Archonix Games	Johan	Battle for Archonia	70,4	82,8	73,6	78,4	77,48
215087038			Group Yenko	Cheryl Schroder	Taxi System	66,6	62,7	77,6	89	75,45
215093585			Group Yenko	Cheryl Schroder	Taxi System	66,6	62,7	77,6	89	75,45
215222539			Group Yenko	Cheryl Schroder	Taxi System	66,6	62,7	77,6	89	75,45
210054689			Simpolygons	Johan	Conquer Africa	71,2	74,5	76,3	76	75,16
215103106			Simpolygons	Johan	Conquer Africa	71,2	74,5	76,3	76	75,16
215075129			IT Buddies	Melissa Makalima	Student Apartment System	55,2	59,9	83,4	88,4	75,03
215211421			IT Buddies	Melissa Makalima	Student Apartment System	55,2	59,9	83,4	88,4	75,03
215140826			Sandstorm	Cheryl Schroder	Number Crumble Educational System	81,2	86,8	72,4	57,6	73,16
215063023			Semi-colon Central	Johan	Rquest	80,4	69,7	73,3	72,6	72,72
215164792			Semi-colon Central	Johan	Rquest	80,4	69,7	73,3	72,6	72,72
215173228			PioneerX Solutions	Cheryl Schroder	Digitally Integrated Voting System	62,2	71,9	69,5	77,4	71,86
215222938			PioneerX Solutions	Cheryl Schroder	Digitally Integrated Voting System	62,2	71,9	69,5	77,4	71,86
215258460			PioneerX Solutions	Cheryl Schroder	Digitally Integrated Voting System	62,2	71,9	69,5	77,4	71,86
215358856			PioneerX Solutions	Cheryl Schroder	Digitally Integrated Voting System	62,2	71,9	69,5	77,4	71,86
215195566			Semi-colon Central	Johan	Rquest	80,4	69,7	69,8	72,6	71,67
213444844			Global-Soft	Lynn Futcher	NMMU Student2Student System	62,8	55,4	74,5	83,8	70,39
215078225			Global-Soft	Lynn Futcher	NMMU Student2Student System	62,8	55,4	74,5	83,8	70,39

Appendix B: Prediction Model Code

Artificial Neural Network Code

```
# DATA PREPROCESSING
# Import libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Quater 3.csv')
X = dataset.iloc[:, 0:10].values
y = dataset.iloc[:, 10].values

# Encoding categorical data
# Not necessary in this case

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size = 0.5)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# CREATE THE ANN
# Importing the Keras libraries and packages
import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.wrappers.scikit_learn import KerasClassifier

# Fit and evaluate the ANN
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score

def build_classifier():
    classifier = Sequential()
    classifier.add(Dense(units = 5, init = 'uniform', activation =
'relu', input_dim = 10))
    classifier.add(Dropout(p = 0.2))
    classifier.add(Dense(units = 5, init = 'uniform', activation =
'relu'))
    classifier.add(Dropout(p = 0.2))
    classifier.add(Dense(units = 1, init = 'uniform', activation =
'sigmoid'))
```

```
    classifier.compile(optimizer = 'adam', loss =
'binary_crossentropy', metrics = ['accuracy'])
    return classifier
classifier = KerasClassifier(build_fn = build_classifier,
batch_size = 10, epochs = 500)
accuracies = cross_val_score(estimator = classifier, X = X_train,
y = y_train, cv = 10, n_jobs = 1)
mean = accuracies.mean()
variance = accuracies.std()
precision = cross_val_score(classifier, X_train, y_train, cv=10,
scoring='precision')
meanPrecision = precision.mean()
precisionVar = precision.std()
recall = cross_val_score(classifier, X_train, y_train, cv=10,
scoring='recall')
meanRecall = recall.mean()
recallVar = recall.std()
f1 = cross_val_score(classifier, X_train, y_train, cv=10,
scoring='f1')
meanF1 = f1.mean()
fVar = f1.std()
```

Decision Tree Code

```
# DATA PREPROCESSING
# Import libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Quater 3.csv')
X = dataset.iloc[:, 0:10].values
y = dataset.iloc[:, 10].values

# Encoding categorical data
# Not necessary in this case

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size = 0.5)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# CREATE THE DT
# Importing the Keras libraries and packages
import sklearn.datasets as datasets
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
```

```
# Fit and evaluate the DT
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score

dt = DecisionTreeClassifier()

dt.fit(X_train, y_train)

# TEST THE ANN
# Predicting the Test set results
y_pred = dt.predict(X_test)
y_pred = (y_pred > 0.7)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

# Get the accuracy score
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)*100

accuracies = cross_val_score(dt, X_train, y_train, cv=10)
mean = accuracies.mean()
variance = accuracies.std()
precision = cross_val_score(dt, X_train, y_train, cv=10,
scoring='precision')
meanPrecision = precision.mean()
precisionVar = precision.std()
recall = cross_val_score(dt, X_train, y_train, cv=10,
scoring='recall')
meanRecall = recall.mean()
recallVar = recall.std()
f1 = cross_val_score(dt, X_train, y_train, cv=100, scoring='f1')
meanF1 = f1.mean()
fVar = f1.std()
```

Naïve Bayes Classifier Code

```
# DATA PRE-PROCESSING
# Import libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB, BernoulliNB,
MultinomialNB

# Importing the dataset
dataset = pd.read_csv('Quater 3.csv')
X = dataset.iloc[:, 0:10].values
y = dataset.iloc[:, 10].values

# Splitting the dataset into the Training set and Test set
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size = 0.5)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# Instantiate the classifier
gnb = GaussianNB()

# Train classifier
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
y_pred = (y_pred > 0.7)

# Making the Confusion Matrix
# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

# Get the accuracy score
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)*100

# Fit and evaluate the DT
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score

accuracies = cross_val_score(dt, X_train, y_train, cv=10)
mean = accuracies.mean()
variance = accuracies.std()
precision = cross_val_score(dt, X_train, y_train, cv=10,
scoring='precision')
meanPrecision = precision.mean()
precisionVar = precision.std()
recall = cross_val_score(dt, X_train, y_train, cv=10,
scoring='recall')
meanRecall = recall.mean()
recallVar = recall.std()
f1 = cross_val_score(dt, X_train, y_train, cv=100, scoring='f1')
meanF1 = f1.mean()
fVar = f1.std()
```