EARLY PREDICTION OF STUDENTS AT RISK IN A VIRTUAL LEARNING ENVIRONMENT USING ENSEMBLE MACHINE LEARNING TECHNIQUES

By

RANJIN SOOBRAMONEY

(21855873)

Submitted in fulfilment of the requirements for the Degree of

MASTER OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

in the

DEPARTMENT OF INFORMATION TECHNOLOGY

in the

FACULTY OF ACCOUNTING AND INFORMATICS

December 2021

Declaration

I, Ranjin Soobramoney, declare that this dissertation is a representation of my own work both in conception and execution. This work has not been submitted in any form for another degree at any university or institution of higher learning. All information cited from published or unpublished works has been acknowledged.

13 December 2021

Ranjin Soobramoney Date

Approved for final submission

Supervisor 13 December 2021

Dr Alveen Singh Date

Acknowledgements

The greatest of appreciation goes to my supervisor Dr Alveen Singh. Your patience, professionalism and most of all, your dedication to your students did not go unnoticed.

I would also like to acknowledge my family and friends for their encouragement and support throughout this sometimes, challenging journey.

I would like to thank my HOD, Dr Jeanette Wing for all her support and encouragement.

A special thanks to Professor Sunday Ojo for his assistance in helping to finalize this dissertation.

Finally, to my colleague and friend Sumaya Hoosen. Thank you for your constant support and encouragement throughout this research journey.

Dedication

To my mum for being the person you are.

Abstract

Students at risk (SAR) are those students who are considered to have a higher probability of failing academically or dropping out of an academic programme. The literature reveals that SAR is a global problem at Higher Education Institutions (HEIs). A high failure rate can not only harm the reputation of the HEIs, but if left unchecked, can be detrimental to these HEIs. The problem of identifying SAR is a pervasive and persistent one. However, early identification of SAR will allow for timely and focused interventions, thereby reducing the problem. Various techniques have been used by HEIs to identify SAR. The traditional statistical approach is one such technique. One of the key challenges with this technique however, is that it often requires a large amount of manual analysis of the data to predict SAR, which in turn also makes early predictions of SAR more computationally challenging. To overcome some of the challenges of the traditional statistical approach, machine learning-based techniques have been proffered to predict SAR. Since machine learning (ML) models are based on the input data rather than the underlying problem, they are expected to have better predictive capabilities than traditional statistical models. Several ML-based techniques have been applied to predict SAR with varying degrees of success. This study proposes the use of ensemble ML techniques for early and accurate prediction of SAR using students' demographic and weekly online Virtual Learning Environment (VLE) data. Aggregating the predictions of a group of ML classifiers is expected to provide a better generalization performance than each of the individual classifiers on their own. The use of ensemble ML techniques for this study will provide an improved solution to the problem of predicting SAR. To this end, this study focused on training forty different ML predictive models, one for each week of the semester, using twenty-five different ML classifiers. Each model was trained using students' demographic data combined with data from their weekly interactions with a VLE. Based on the training results, four classifiers, namely AdaBoostClassifier, LGBMClassifier, RandomForestClassifier, and XGBClassifier were selected as base learners for the ensemble classifier. Hyperparameter optimization was performed using Random Search on each of the four classifiers. These classifiers were then used to create a voting classifier ensemble for each of the forty weeks, with 10-fold cross validation being used to evaluate the predictive models. The results show that the voting classifier ensemble method outperformed the individual classifiers overall over forty weeks and can thus provide an improved solution to the problem of predicting SAR.

Keywords: Students at Risk, Ensemble learning, Lazypredict, Machine Learning Algorithms, Virtual Learning Environment.

List of Publications

Soobramoney, R., & Singh, A. (2019, 6-8 March 2019). Identifying Students At-Risk with an Ensemble of Machine Learning Algorithms. Conference Proceedings: 2019 Conference on Information Communications Technology and Society (ICTAS). Indexed by IEEE Explore.

Table of Contents

List of Tables

List of Figures

List of Code Snippets

Abbreviations

ANN	Artificial Neural Networks
DHET	Department of Higher Education and Training
DT	Decision Tree
EDM	Educational Data Mining
HE	Higher Education
HEI	Higher Education Institution
kNN	K-Nearest Neighbor
LMS	Learning Management System
LR	Logistic Regression
ML	Machine Learning
MLA	Machine Learning Algorithm
NB	Naïve Bayes
OULAD	Open Learning University Analytics Dataset
RF	Random Forest
RO	Research Objective
RO	Research Objective
SAR	Students at Risk
SVM	Support Vector Machine
VLE	Virtual Learning Environment

Chapter 1: Introduction

This chapter presents an outline of this dissertation. It begins by providing a background to the problem of students at risk (SAR) and the need for their early identification. The research problem, followed by the research aims and objectives, the research process, the significance as well as the contribution of the study, are then presented. The chapter concludes with an outline of how the rest of the dissertation is organized.

1.1 Background

The literature reveals the SAR problem as a global problem at HEIs. Concerns over SAR are not new. Some time ago, one of the most cited commentators on student retention and dropouts stressed that:

"*Despite all the research that has been conducted to date, little work has been devoted to the development of a model of student persistence that would provide guidelines to institutions for creating policies, practices and programs to enhance student success*"

(Tinto, 2005).

Studies have shown that students are at risk for various reasons. However, early identification of SAR will allow for timely and focused interventions, thereby reducing the problem of SAR. In the context of this study, SAR are those considered to have a higher probability of failing academically or dropping out of an academic programme. This dissertation investigates the feasibility of using ensemble ML techniques for early and accurate prediction of SAR at a HEI using students' demographic and weekly online VLE data. This in turn contributes towards identifying and providing support for SAR as early as possible.

1.2 Research Problem

Various techniques have been used by HEIs the world over to review the performance of students in order to predict SAR. The traditional statistical approach is one such technique. One of the key challenges with this technique is when there is a large number of courses and students, and it becomes somewhat impractical to review students' performances in a timely and frequent manner across all courses (Fan, Han, & Liu, 2014; Ming, Xueshuai, & Qian, 2019; Xing, Chen, Stein, & Marcinkowski, 2016). This approach often requires a large amount of manual analysis of the data to predict SAR, which in turn also makes early predictions of SAR more computationally challenging (Adnan et al., 2021). Shah (2016) suggested that the traditional statistical technique used to identify SAR is more suited to low-dimensional datasets. Rovira, Puertas, and Igual (2017) argued that traditional statistical models are

primarily based on assumptions about the underlying problem and incorrect assumptions about the underlying problem may result in poor predictions.

To overcome some of the challenges of the traditional statistical approach, ML-based techniques have been proffered to predict SAR. Rovira et al. (2017) suggest that since ML models are based on the input data rather than the underlying problem, they are expected to have better predictive capabilities than traditional statistical models. The authors also suggest that ML models can easily adapt to new data while the underlying assumptions of statistical models become obsolete when the data changes over time.

Several current ML-based studies such as those by (Adnan et al., 2021; Al-Azawei & Al-Masoudy, 2020; Aljohani, Fayoumi, & Hassan, 2019; Azizah, Pujianto, Nugraha, & Darusalam, 2018; Chui, Fung, Lytras, & Lam, 2020; Haiyang, Wang, Benachour, & Tubman, 2018; Hassan et al., 2019; Heuer & Breiter, 2018; Hlosta, Zdrahal, & Zendulka, 2017; Hussain, Zhu, Zhang, & Abidi, 2018a; Jha, Ghergulescu, & Moldovan, 2019; Rizvi, Rienties, & Khoja, 2019; Wasif, Waheed, Aljohani, & Hassan, 2019) were used to identify SAR in a VLE. It was found that while some of these solutions had good predictive performance, they were not without limitations. Of these studies, none seem to have performed any hyperparameter optimization nor used any ensemble ML techniques to improve predictive performance.

This study thus proposes the use of ensemble ML techniques for early and accurate prediction of SAR using students' demographic and weekly online VLE data. Aggregating the predictions of a group of classifiers is expected to provide a better generalization performance than each of the individual classifiers on their own (Beyeler, 2017; Géron, 2019; Raschka & Mirjalili, 2017). Instead of relying on a single expert (classifier), predictions of individual experts (classifiers) can be strategically combined to come up with a more accurate prediction (Polikar, 2012).

The use of ensemble ML techniques for this study will provide an improved solution to the problem of predicting SAR.

1.3 Research Question, Aim and Objectives

This section presents the research question, aim and objectives of this study.

Research Question: What ML ensemble can be used for early and accurate prediction of SAR using students' demographic and weekly online Virtual Learning Environment (VLE) data?

Research Aim: The main aim of this study is to employ ML-based predictive modelling techniques in early and accurate prediction of SAR at a HEI using students' demographic and weekly online VLE data.

To answer the research question, the following research objectives (ROs) were formulated.

Research Objectives:

RO1: To conduct a comparative performance analysis of MLAs used in predicting SAR.

RO2: To develop a MLA-based SAR prediction model using an ensemble of best-performing MLAs.

RO3: To experimentally validate this model using demographic and weekly online VLE data.

1.4 Overview of the Research Methodology

To achieve the research aim, this study followed a quantitative research strategy, using a post-positivism paradigm. To achieve the research objectives, the methodology of this study consisted of two broad phases.

Phase one addressed RO1. An extensive review of the literature highlighted the impact of SAR. Some of the main challenges of using traditional statistical approaches to identify SAR were also identified through the literature review. Based on the challenges and shortcomings of the traditional statistical approaches, a ML approach was proposed as an alternative for the improved identification of SAR. The literature review also provided initial information on some of the characteristics that could be used to identify SAR. A further review of the literature that used ML-based approaches to predict SAR in a VLE, enabled this researcher to identify certain limitations and shortcomings. This researcher was able to address some of these

limitations by developing a MLA-based SAR prediction model using an ensemble of best-performing MLAs for early and accurate prediction of SAR.

Phase two addressed RO2 and RO3. The quality and format of the raw student data is important to enable the ML models to accurately identify SAR. The raw data requires pre-processing so it can be transformed into a form that can be used for ML. In real-life situations, the data can be quite messy and may thus require a lot of pre-processing before it can be used in ML. A rigorous search was conducted for an appropriate dataset which was located and extensively pre-processed for ML. The pre-processing stage entailed, but was not limited to, performing an exploratory analysis of the data, deleting or updating rows (or columns) that contained missing data, converting all non-numeric data to numeric data, scaling the data, and engineering new features or attributes from existing student attributes. ML metrics such as accuracy, balanced accuracy, F1 score and ROC AUC score, were discussed in the context of their use in evaluating the performance of the MLAs. The pre-processed data was used as input to the ML models in order to identify the best performing classifiers to be used as base classifiers for further improvements. The ML modelling process of this study to determine the best performing base classifiers entailed modelling and prediction of SAR on a weekly basis for forty weeks of a semester. This entailed training forty different ML predictive models, one for each week of the semester, using twenty-five different MLAs. Each model was trained using students' demographic data, combined with data from their weekly interactions with a VLE. For each week, starting at week 0, the model was supplied with students' demographic data (which remained the same for each of the forty weeks), together with data from the students' VLE interaction from all of the preceding weeks, up to and including the current week. For instance, for week 2, VLE training data was supplied to the model as data from week 0, week 1, and week 2. This process was repeated for each of the forty weeks. The predictive capability of each model was then evaluated using the four different performance metrics, namely accuracy, F1 score, and ROC AUC score.

The MLAs with the highest scores for each of the different metrics (accuracy, balanced accuracy, F1 Score, and ROC AUC score), were identified for each of the forty weeks. The best-performing MLAs were then selected as base classifiers. Hyperparameter optimization was done using Random Search to tune appropriate hyperparameters of the four base classifiers, namely LGBMClassifier, AdaBoostClassifier, RandomForestClassifier and XGBClassifier. A soft voting ensemble classifier was then used to combine the four classifiers

in order to create an improved classifier with better prediction accuracy than any of the four individual classifiers.

1.5 Contribution of the Study

Given the perennial issue of SAR and its consequent negative impact on HEIs, any interventions that can help to minimize the SAR problem would be welcome by the HEIs.

The intended contribution of this study towards employing ML-based predictive modelling techniques in early and accurate prediction of SAR at a HEI are enunciated as follows:

a) Contribution to theory: This contribution is derived from answering the research question "What ML ensemble can be used for early and accurate prediction of SAR using students' demographic and weekly online Virtual Learning Environment (VLE) data?". Through the construction of a validated MLA-based SAR predictive model, this study found that a soft voting ensemble of four MLAs, namely AdaBoostClassifier, LGBMClassifier, RandomForestClassifier, and XGBClassifier, can be used for early and accurate prediction of SAR using students' demographic and weekly online VLE data.
b) Contribution to methodology: The validated methodology of this study can be reused by other researchers.
c) Contribution to practice: The results of this study can be used in practice for the early prediction of SAR.

1.6 Organization of the Dissertation

This dissertation is organized into six chapters as described below:

Chapter 1: Introduction

This chapter begins by providing some background to the problem of SAR, and the need for early identification of SAR: the research problem, followed by the research aims and objectives, the research process, the significance, as well as the contribution of the study. The chapter concludes with an outline of how the rest of the dissertation is organized.

Chapter 2: Literature Review

This chapter provides a review of the literature with regards to SAR: why it is a problem as well as current attempts to alleviate this problem. The first section defines SAR in the context of this study and explores its impact on the global and South African (RSA) HEI landscapes.

The second section provides an exposé of recent attempts to try and alleviate the problem of SAR. The third section posits and provides motivation for the area of ML as a potential candidate in attempting to alleviate the problem of SAR. The chapter concludes with a summary of the findings from the literature.

Chapter 3: A Description of the Research Framework

This chapter presents the components of the research framework which work towards achieving the research objectives. This is followed by a detailed description of the methodology commissioned for the development of a ML-predictive model to identify SAR. The choice of metrics and ML classification techniques used to evaluate the effectiveness of the ML model is discussed.

Chapter 4: A Discussion on Conducting the Experiment

This chapter presents the results of the experiments that used ML techniques to identify SAR. The first part discusses the dataset used and the rationale for its use. The second part discusses the data preparation process, which is a necessary step before the data can be used for ML. This part delves into the details of data cleaning, feature selection and engineering, as well as feature scaling and selection. The ML modelling process of this study entailed modelling and prediction of SAR on a weekly basis for forty weeks of a semester. This entailed training forty different ML predictive models, one for each week of the semester, using twenty-five different MLAs. Different classification metrics used in order to evaluate the performance of the MLAs to identify SAR, will also be discussed.

Chapter 5: A Discussion on the Results of the Experiment

This chapter discusses the findings of the experiments that used ML techniques to identify SAR. Some conclusions based on the findings will also be discussed. The supervised ML methodology as outlined in section 3.4 was followed to identify SAR. The MLAs that were identified from the experiments as suitable candidates to identify SAR, will also be discussed.

Chapter 6: Conclusion, Limitations and Future Research

This chapter will revisit the research objectives stated in Chapter 1. The research question will be answered. The limitations of this study will be discussed together with recommendations to advance the study further. Some clarity will be provided on the contribution of this study towards early identification of SAR.

Chapter 2: Literature Review

2.1 Introduction

The purpose of this chapter is to outline a review of the current literature for this study. As a broad overview, this chapter: contextualizes SAR and then reviews related literature on SAR; tries to show why it is a persistent and complex problem; and evaluates recent attempts to address this problem. The chapter then introduces ML in context of this study and provides an overview of its operation and the part it plays towards a solution to the SAR problem.

Specifically, the first section (2.2) defines SAR in the context of this study and explores its impact on the global and RSA HE landscape. This section seeks to unearth the greater economic impact of SAR on both the RSA economy as well as the HEIs. The second section (2.3) provides an exposé of recent attempts to try and alleviate the problem of SAR. The third section (2.4) posits and provides motivation for the area of ML as a potential candidate in attempting to alleviate the problem of SAR. The chapter ends with a summary of the findings from the literature.

2.2 Students at Risk

The following sections provide an overview of SAR.

2.2.1 Students at Risk Definitions

The Glossary of Education Reform defines SAR as "students or groups of students who are considered to have a higher probability of failing academically or dropping out" (The Glossary of Educational Reform, 2013). The Institute of Education Sciences define SAR as those "students considered in danger of not graduating, being promoted, or meeting other education-related goals" (Institute of Education Sciences, 2008).

In the context of this study, SAR are those students who are in danger of failing a course. Subsequent intervention strategies may be required for these students to succeed in their course (Veerasamy, D'Souza, Apiola, Laakso, & Salakoski, 2020).

2.2.2 Students at Risk in the South African Context

The average graduation rate in 2019 at public HEIs in South Africa was 20.6%, representing a graduation rate of 19.1% for male students and 21.7% for female students (DHET, 2021b, p. 69). The low graduation rate which in turn impacts the throughput rates together with a decline in funding to HEIs (Qonde, 2021) increases the urgent need of HEIs to minimise SAR. The

graduation rate is the ratio of the number of students who graduated in a particular year divided by the total number of students enrolled at HEIs in that year. The throughput rate refers to a cohort of students that complete their qualification within the minimum timeframe set for that qualification (DHET, 2021c).

A democratic South Africa is less than three decades old and considered a developing country and a key member state of the African Union. The performance and attrition rate of tertiary students cannot be fully explored without considering the socio-economic historical context of South Africa. There are several demographic factors stemming from this history that could have a direct or indirect impact on a student's performance at HEIs. Some factors can include the financial circumstances of the student, their living conditions, their parents' level of education, their previous quality of education and the area in which they live (Radunzel, 2017; Soobramoney & Singh, 2019). These factors and possibly several more will need to be considered in order to more precisely identify SAR in this context. It is not always practical to consider all these external factors when trying to identify SAR using traditional statistical or manual methods. This is because these methods primarily rely on making inferences from the data as opposed to finding meaningful patterns in the data (Bzdok, Altman, & Krzywinski, 2018). Simply put, the traditional methods are not designed to find underlying patterns in the data that may impact on determining whether a student is at risk or not.

2.2.3 The Impact of Students at Risk

The HEI landscape is evolving at a rapid pace. There is an ever-increasing demand for access to HE. The number of students enrolled in public HEIs in South Africa increased by 28.3% over the period 2009-2019 (DHET, 2021a, p. 9). An increase in student numbers without a commensurate increase in HE funding is adding to the pressures faced by HEIs. These institutions are expected to meet greater educational targets with fewer resources.

A report by Africheck (2016) indicates that the SA government's share of the funding income to public universities and other tertiary education institutions declined from 49% in 2000 to 39% in 2015. Furthermore, the state subsidy to HEIs has been steadily declining with annual increases less than the Consumer Price Index (Stellenbosch University, 2017). The budget for HEIs has been cut by just over R20 billion for the 2020/2021 period (Qonde, 2021).

HEIs are increasingly under pressure to reduce their dropout rates and increase their throughput rates (Cook & Pullaro, 2010; Gold & Albert, 2006; Government Technical Advisory Centre, 2016). Dropout commonly refers to the withdrawal from an academic programme before

completion. Students who are at risk of failing are likely to contribute to the high dropout and subsequently low throughput rates. This in turn adds additional financial strain on already underfunded HEIs.

Therefore, given the aforementioned statistics, if left unchecked, the SAR problem can be detrimental for the long-term sustainability of HEIs.

South Africa is considered a developing nation and a key member of the African development community. The long-term sustainability and growth of this economy is pivotal in supporting its own strategic objectives and that of the African continent. A skilled workforce and citizenry are a hallmark of any developed nation and a key national imperative as outlined in many government publications. Investment in HE graduates is the bedrock to growth and sustainability of the greater economy (Lusigi, 2019; Pouris & Inglesi-Lotz, 2014)

A recent decision by the South African Government, is to fund education for certain categories of students, using the household income threshold to determine which students receive funding. This decision has contributed to an increase in government spending on student funds. In 2021, more than R43 billion was set aside for this fund (Mofokeng, 2021). The average cost to the government to fund a single HEI student for a year in 2018 was R88 600 (DHET, 2021b). The sustainability of this return-on-investment type funding model is dependent on adequate throughput and low dropout rates. What is noteworthy, is that the student who is funded becomes liable or *must pay back* a portion of their received funding, should they exceed the maximum duration of the qualification. In such a situation, the HEI has produced a graduate that is in debt before they even begin their career.

In 2019, the throughput rate of the cohort of students who entered HEIs in 2012 was 56.4% (DHET, 2020). Thus, over 40% of students from this cohort may never graduate. This statistic points to an alarming loss of investment in these students through bursaries and subsidies. Additionally, it is also a cost in terms of human resources, if large numbers of students leave the HEIs without a qualification. Furthermore, the longer a student takes to graduate, the less they will be contributing to a developing economy, denying the same economy of much-needed skills and entrepreneurship.

Given that South Africa is a developing country with an underperforming economy, the country can ill afford to continue with this huge amount of funding, while the failure rate of students is so high.

If left unchecked, the problem of SAR can have a detrimental impact on the affected students. Student attrition is unfavourable for most students, in the sense that failure to complete their studies may result in these students losing out on both social and economic benefits (Spence, 2012). Students who are educationally, socially and financially disadvantaged are at risk of failing and dropping out of HEIs (Coates, 2005). Najimi, Sharifirad, Amini, and Meftagh (2013) surmised that failing students are susceptible to personal, social, as well as psychological problems, which may result in these students not achieving their educational goals. The authors further point out that failing students may also suffer financial and time losses.

Earlier studies found a strong positive correlation between academic failure and dropping out with increased use of drugs and alcohol (Younge, Oetting, & Deffenbacher, 1996), and more alarmingly, that academic failure was one of the most common reasons for student suicide (Meilman, Pattis, & Kraus-Zeilmann, 1994). Those characteristics have not changed much in the present day as confirmed by: (Ajjawi, Dracup, Zacharias, Bennett, & Boud, 2020; Pillay, 2021; Zamayirha, 2018).

Berens, Schneider, Görtz, Oster, and Burghoff (2018) posited that student attrition had a negative impact on the student, HEIs, and the general public, while any gains made by the student were offset by the wasted public resources when they drop out. The authors further suspected that in order to reduce student attrition rates, it was important to identify not only students that were at risk of dropping out, but also to determine what the underlying factors were leading to them dropping out.

Besides the financial losses incurred when students drop out, Larsen, Sommersel, and Larsen (2013) also suggest that those students may feel inadequate and this may possibly lead to them being shunned by society. Three major economic stakeholders are highlighted in HE, namely students and often their families, HEIs, and government (Berens et al., 2018; Larsen et al., 2013). Each of these stakeholders will have different concerns about students dropping out. Since dropping out will negatively impact all three stakeholders, pre-emptively identifying SAR can help to reduce the dropout rate to the benefit of all the stakeholders. However, should there not be early interventions to help SAR, it could hamper the progress of these students in terms of completing their course in the minimum amount of time. This will in turn have cost implications for the student, the HEI, and the country. Academic failure and subsequent

dropping out cannot be seen as affecting the individual student in isolation, but also society as a whole.

2.3 Identifying Students at Risk: A Review of Some Current Approaches

The literature shows that the SAR problem is common all over the world and students are at risk for various reasons. Tinto (1975) produced one of the earliest, and what was considered a highly influential, theoretical explanation of student attrition in his path analysis model. The model suggested that both students' social and academic influence determined completion of their studies. It identified students' family background, personal characteristics, previous schooling, prior academic performance, and interactions between students and staff, as factors influencing the success or failure of students at HEIs. That study is based on the premise that determining the best set of predictor factors to identify SAR, will enable more appropriate intervention strategies to help these students. HEIs the world over have used various approaches and techniques to monitor the performance of students in order to identify SAR. There have been varying degrees of success in this regard. Some of the approaches are highlighted in the next sub-sections.

2.3.1 Traditional Statistical Approaches to Identify Students at Risk

The literature describes various approaches that have been used to identify SAR. The traditional statistical approach is one such approach, which has achieved varying degrees of success. Using the traditional statistical approach to identify SAR often involves descriptive or inferential statistics. Descriptive statistics is used to describe and summarize the data in a dataset, while inferential statistics involve analysing input data using statistical models and assumptions to make inferences about the data from a given dataset (Bhandari, 2021; Sarka, 2021). Thus, with inferential statistics, input data is supplied to a statistical model to draw conclusions about a larger population. Below are some studies that used traditional statistical methods to identify SAR.

Najimi et al. (2013) investigated the factors that may contribute to students' failure at the Isfahan University of Medical Sciences in Iran. Demographic data and study information of 280 students were collected via a questionnaire. The data was analysed using the SPSS statistical software. An analysis of the results showed that curriculum factors related to the educator, the learning environment, family, and socioeconomic factors, had a key impact on poor student performance. The study was a descriptive, qualitative study and the risk factors that were identified were actually from the students' point of view.

Smith, Therry, and Whale (2012) attempted to build a predictive model using two statistical methods, namely linear discriminant analysis and logistic regression, in order to identify SAR on a first-year accounting course at an Australian HEI. The prediction results never exceeded 67%. The authors concluded from the results obtained that they would not be able to use the findings to make generalized predictions.

Milne, Jeffrey, Suddaby, and Higgins (2012) closely examined the relationship between the use of a learning management system (LMS), such as Moodle, and the performance of students. A statistical analysis was done to compare students who passed the course with their LMS activity in the first week of lectures. The results of the analysis could not draw any definitive conclusions between the use of LMS and the success rate of students. This could point to the possibility that other factors needed to be considered that were absent from the study.

Radunzel (2017) used statistical modelling to try and identify SAR after their first year of study. A wide range of characteristics was used, such as travel distance from the HEIs and demographics, including gender, ethnicity, and parents' educational level. Hierarchical multinomial regression models were used to predict dropping out. The findings of the study suggested that many different factors, both academic and non-academic, were important to predict student attrition. The demographics of students, such as their socio-economic status and even their parents' educational level, were found to be factors that played a part in the retention rates of students. The author also suggested that students that come from lower-income households are more likely to have other non-academic obligations, such as needing to work to supplement the family income or even other family responsibilities, that could impact on their efforts at a HEI.

Shaw and Mattern (2013) investigated the factors that may determine which students drop out of a consortium of HEIs in the USA. The study found that academic performance as well as demographics, such as gender, race, ethnicity, and parental level of education of the student, played a major role in student retention.

Rajandran et al. (2014) examined the factors that affect academic performance from a qualitative point of view. Cross-tabulation and multinomial logistic regression were used to analyse a sample of 100 students at the University of Malaysia. The review of the literature by the authors found a lot of contradictions with regards to the factors that affect student performance, despite similar methods such as cross-tabulation and logistic regression being

used. The qualitative nature with statistical analysis of the data could be a reason for these inconsistencies.

Oyerinde and Chia (2017) attempted to predict student performance using learning analytics with multiple linear regression. Learning analytics is an integration of data analytics and data mining techniques. The study was unable to identify which student attributes contributed to better academic performance.

Urrutia-Aguilar et al. (2016) used a logistic regression model to predict the performance of first-year medical students at the University of Mexico. The authors indicated the use of academic, psychological, and vocational variables. This was an observational and descriptive study consisting of 1205 students completing questionnaires with 925 valid responses. The accuracy rate for the prediction was 77%.

Thompson, Li, and Shulruf (2019) used a statistical procedure called discriminant function analysis (DFA) to classify SAR from other students in a group of 700 medical students at the University of New South Wales in Australia. Prior academic achievement and interview scores were used. The classification accuracy was 73.7%

From the literature that was reviewed, in most cases, it appears that the authors could not draw definitive conclusions from the results. Reasons proffered varied from low prediction rates of the study (Smith et al., 2012; Thompson et al., 2019), to contradictions in the possible factors that contribute to identifying SAR (Rajandran et al., 2014), and the qualitative and descriptive nature of the study (Milne et al., 2012; Najimi et al., 2013; Urrutia-Aguilar et al., 2016). These reasons and others are elaborated upon in the next sub-section.

2.3.2 A Summary of the Challenges of the Traditional Statistical Approaches to Identify Students at Risk

Despite HEIs expending a lot of resources on trying to identify SAR, the manual or traditional statistical means of identifying students at risk is problematic because by the time a SAR is identified using traditional means, and it may be too late to help the student who may then be in danger of dropping out.

Despite interventions by HEIs to address the problems of SAR, there has not been any significant improvements in throughput or reduction in dropout rates. This trend also seems to

have been noted by (Tinto, 2012) in other countries, such as the USA. This brings to the fore a need to investigate the effectiveness of the current techniques being used to identify SAR.

One of the key challenges with the traditional statistical approach to identifying SAR, is when there are a large number of courses and students, it becomes somewhat impractical to review students' performances in a timely and frequent manner across all courses. Academic staff who are tasked with attempting to identify SAR, often tend to rely mainly on assessment marks as the primary source of information. For practical reasons, there may be several other underlying factors that could be overlooked when attempting to identify SAR. Shah (2016) suggests that the traditional statistical modelling techniques is more suited to low-dimensional datasets. Furthermore, early identification and intervention procedures for SAR should not rely solely on assessment marks because very often, the first assessment only happens at least a few weeks into the semester.

De Villiers and Farrington (2019) attempted to predict SAR in a first-year accounting course at the Nelson Mandela Metropolitan University in South Africa. The authors used demographic and educational historical data for analysis. The methodology used was descriptive statistics and discriminant analysis. In reviewing the literature, the authors identified several other studies which contradicted each other with regards to the actual demographic and educational variables that influenced student performance. This may point to another shortcoming of the traditional statistical methods to identify SAR. The study further found that even with improved statistical analysis, the factors were *manually* chosen to identify SAR based on the literature. Other factors that could have played a more significant role in improving the predictive accuracy of identifying SAR were left out owing to the manual nature of the process of selecting the factors.

In essence, a timely and more accurate prediction of SAR will allow for more meaningful interventions to assist these students. This may subsequently help reduce the high dropout rate of students. These earlier interventions and remedial actions may thus obviate the need for much costlier interventions at a later stage and minimize the broader socio-economic impact as outlined in earlier sub-sections. Identifying SAR very often involves manually analysing their prior assessment results. While prior assessment results may be a major contributor to identifying students at-risk, several underlying factors may be overlooked that can have an impact on the performance of a student. Intuitively, factors such as the financial status of the student, the level of education of his or her parents, the quality of the educational development

of the school the student has attended, and perhaps even the area in which the student lives, can have varying levels of impact on a student's performance. Often when dealing with a large number of students of varying backgrounds and circumstances, manually trying to identify SAR can be both time-consuming and often inaccurate. It is neither feasible nor practical to consider all these factors (and perhaps several more) when manually trying to identify SAR. As a consequence, manually trying to identify SAR is often primarily based on prior academic or school results, and very little else. Few, if any, other factors are taken into account.

Furthermore, it may also be neither feasible nor practical to analyse large amounts of data using manual methods in order to identify SAR. Often the methodologies that are used end up being qualitative and mainly descriptive in nature. Even when traditional statistical methods are used, they have a limitation in that they are time invariant and static in nature.

Rovira et al. (2017) argued that traditional statistical models are primarily based on assumptions about the underlying problem. Thus, incorrect assumptions about the underlying problem may result in poor predictions. Since ML models are based on the input data rather than the underlying problem, they are expected to have better predictive capabilities than traditional statistical models. The authors also suggest that ML models can easily adapt to new data while the underlying assumptions of statistical models become obsolete when the data changes over time.

At this juncture, this study considers the use of ML models that could provide a genesis towards overcoming both the limitations of the manual method, and the time-invariant, static nature of traditional statistical methods. This in turn is expected to produce better predictive results. This could be especially important when it comes to student retention (as a means to prevent dropping out). Tinto (1987) argued that student retention was not an immediate once-off event, but rather an ongoing process. The dynamic nature of the process of student retention and identifying SAR will lead to more effective and relevant intervention strategies.

This study adopts the stance that classification techniques using an ensemble of MLAs have promising prospects to more effectively predict SAR, by including several factors that may often not even be practical with manual methods or even traditional statistical methods. These models in the least, can complement the manual methods. SAR can be identified much sooner and with possibly greater accuracy. This in turn contributes towards identifying and providing support for SAR as early as possible.

2.3.3 A Review of Machine Learning Methods for Prediction of Students at Risk

Given the growing demands on HEIs, a manual only or traditional statistical means of identifying SAR can be time-consuming and often ineffective. More importantly, given the numerous and complex factors that can impact on a student's performance, a timely and more accurate prediction of SAR is required. This could allow for more robust interventions to help reduce the high dropout rate of students. These earlier interventions and remedial actions may thus obviate the need for much costlier interventions at a later stage.

ML seems a worthy candidate for predicting student academic performance in order to identify SAR. Despite recent research into the application of MLAs in the greater HEIs problem domain, these are not without deficiency and there is room for improvement.

The concept of learning is to obtain a knowledge of general concepts by using particular training examples. By being exposed to a subset of a set of items for instance, one may be able to categorize a specific item. For instance, if one were to attempt to identify an orange from a set of fruits, one could have a Boolean function to infer an orange from all other types of fruits. ML refers to the automated detection of meaningful patterns in a set of data without explicit programming.

The literature more often than not attributes the first formal definition of ML to Samuel (1959) as:

"Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed".

However, this appears to be an interpretation of the actual quote by the author, which was:

"Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort"

(Samuel, 1959, p. 1).

A more formal definition of ML by Mitchell (1997, p. 2) is:

> *"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."*

Thus, based on the above premise and in the context of this study:

- Tasks *T* are the decisions that the MLAs need to make to determine SAR.
- Performance measure *P* is the classification accuracy as a percentage in terms of the number of students correctly predicted as SAR.
- Experience *E* is the collection of student data with the given classifications of SAR and not SAR, and the process of using that data.

2.3.4 Machine Learning Methods

Two broad types of ML methods were considered for this study namely, supervised and unsupervised.

In supervised learning, given a training set of data *D* made up of *N* input-output pairs of labelled data of the form $D = \{(\mathrm{x}_i, y_i)\}_{i=1}^{N}$, where *i* =1..*N*, the goal is to learn a mapping from inputs x to outputs *y* (Murphy, 2012, p. 33). This mapping can then be used on unseen data to predict the output. Generally, x_i is a vector of features or attributes, while y_i is either a discreet value from a finite set of values or a continuous value. While both cases involve predicting an output, if y_i is discreet, then it is a classification problem, and if y_i is continuous, then it is a regression problem. Furthermore, for classification problems, if there are only two possible output values, it is regarded as a binary classification problem. If there are more than two possible output values, then it is regarded as a multiclass classification problem.

In unsupervised learning, one is only given the inputs, $D = \{(\mathrm{x}_i)\}_{i=1}^{N}$, where $i = 1..\mathrm{N}$, and the goal is to find similarities or patterns in the input data (Murphy, 2012, p. 33). The data is organised into a group of clusters to describe and model its underlying structure.

This study aims to determine if a student is at risk or not at risk. Since the dataset for this study contained a set of input student attributes that were trained to learn a mapping to a binary output (student is at risk or not at risk), this study became a binary classification problem using supervised ML techniques.

2.3.5 A Brief Overview of Machine Learning Algorithms Used in the Students at Risk Identification Problem

A review of the literature seems largely unsure of any one specific MLA that performs best in every context in order to identify SAR (Soobramoney & Singh, 2019). The *No Free Lunch*

theorem from Wolpert and Macready (1997) argues that there is no single classification algorithm that can outperform every other classification algorithm in every problem area.

Since it may be difficult to answer the question: "Which is the best MLA that can be used to identify SAR?", one solution would be to evaluate all candidate algorithms and choose the one with the best accuracy. Unfortunately, accuracy scores alone may not always be sufficient. For instance, if the dataset has class imbalances where one output class type is much greater than the other, then the MLA would be biased towards that specific class during training, and will thus not be able to generalize and predict correctly for unseen student data (Ali, Salleh, Saedudin, Hussain, & Mushtaq, 2019; Li, Bellotti, & Adams, 2019; Priya & Uthra, 2021). Thus, other ML metrics such as F1 score, balanced accuracy, and ROC AUC score will need to be considered when identifying SAR (Luque, Carrasco, Martín, & de las Heras, 2019; Mathews & Hari, 2019; Prenkaj, Velardi, Stilo, Distante, & Faralli, 2020; Sarkar, Khatedi, Pramanik, & Maiti, 2020), especially if the dataset is imbalanced. These metrics are discussed later in this study.

Linear models (such as Logistic Regression (LR), Naïve Bayes (NB), and Support Vector Machine (SVM)), non-parametric models such as k-nearest neighbor (kNN), Artificial Neural Networks (ANN) (such as multilayer perceptron), tree based models such as decision tree (DT), and ensemble models including random forest (RF) and gradient boosting models (such as XGBoost (XGB), LightGBM (LGBM) and AdaBoost (AB)), were considered for this study. Some of these models are discussed below.

2.3.5.1 Logistic Regression

LR is used to estimate discrete values based on a given set of independent variables. It predicts the probability of the occurrence of an event by fitting data to a logistic function (Patel, Shah, Sanghvi, & Manan, 2020).

2.3.5.2 Artificial Neural Network

ANNs are models that simulate biological neurons of the brain. The ANN has interconnected nodes with an input, output and at least one hidden layer (Bali, Sarkar, & Sharma, 2018) . Activation functions activate certain neurons to get the desired output (Kakarla, Krishnan, & Alla, 2021). The architecture of a simple neural network is shown in Figure 2.1.

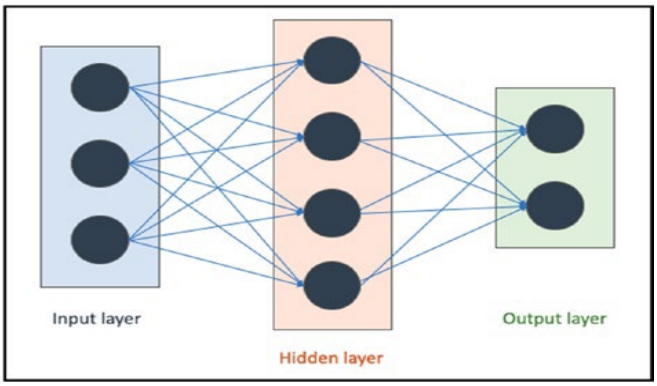

Figure 2.1: Simple neural network architecture. Source: Kakarla et al. (2021)

2.3.5.3 Decision Tree

A DT graphically represents a set of possible outcomes of a decision using "if-then" rules. A path exists with classification rules from the root (or parent) node to the leaf nodes which represent the decisions that can be taken based on the input (Kakarla et al., 2021). The architecture of a credit card approval example is shown in Figure 2.2

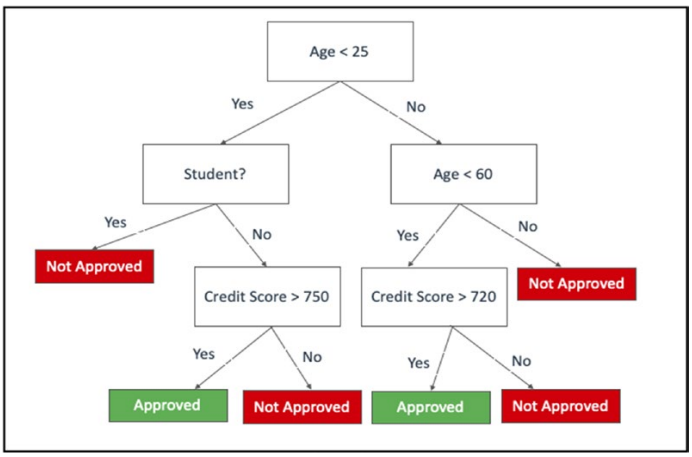

Figure 2.2: Decision tree architecture example. Source: Kakarla et al. (2021).

2.3.5.4 Ensemble Models

Ensemble models typically combine the predictions of individual base ML models and a weighted average or majority voting is used which is expected to improved predictions (Bali et al., 2018; Polikar, 2012). Two major type of methods that are used in ensemble models are bagging and boosting.

Bagging (or bootstrap aggregation), combines the predictions from several base models, using several training samples in parallel, which are randomly generated, in order to make more

accurate predictions (Bali et al., 2018; Polikar, 2012; Zhou, 2012). RF is an example of a commonly used bagging method, in which the RF classifier trains several DTs in parallel using different subsets of training data (Bali et al., 2018; Misra & Li, 2020; Zhou, 2012).

Boosting entails building the ensemble model in an incremental fashion by training each base classifier sequentially in order to learn the instances that were misclassified. Strong ensembles are formed by combining weak learners and training them over multiple iterations, through adjusting the weights of the weaker learners (Bali et al., 2018; Polikar, 2012; Zhou, 2012). Commonly used boosting models are AB, XGB and LGBM.

Ensemble models can be used to improve the performance of MLAs for identifying SAR (Ajibade, Ahmad, & Shamsuddin, 2019; Pandey & Taruna, 2018; Polyzou & Karypis, 2019; Rahman & Islam, 2017; Sahin, 2020). Ensemble learning, which uses ensemble modelling techniques will be discussed in section 2.4.2.

2.4 Machine Learning-Based Models for Predicting Students at Risk

The following sections provide an overview of ML-based models for predicting SAR.

2.4.1 Creating Baseline Machine Learning Models in the Students at Risk Identification Problem using Lazypredict

Various studies employed a number of different MLAs to identify SAR with mixed results. There is no single classification algorithm that can outperform every other classification algorithm in every problem area (Wolpert & Macready, 1997). One possible approach in deciding which MLAs were most effective to identify SAR, was to start with a set of commonly used MLAs and then iterate to establish some benchmarks that could be improved. Another approach that was considered was to start by training as many MLAs as possible and then selecting a subset as baselines for further improvements. However, given that for this study a dataset in excess of 26000 instances needed to be trained for each of forty weeks, the computation time taken to optimally train all the MLAs would have been a major challenge. To resolve this problem, a python wrapper library called lazypredict developed by Pandala (2020) was used to fit and evaluate all ML models from Python's scikit-learn package and several other ML models that are not found in the scikit-learn package, in a relatively shorter computational time. For this study, this included twenty-five MLAs were used to create baseline predictive models which allowed for relatively quick performance benchmarks to be

determined. This in turn enabled a better understanding of the direction to be taken to refine the selected baseline ML models for better prediction results. Since lazypredict is a relatively new library, a review of the literature found three studies that used it. The studies by (Barrionuevo, Ríos, Williams, & Ramos-Grez, 2021; Butcher, 2021; Manafiazar et al., 2021) used lazypredict but none of these studies entailed predicting SAR. Table 2.1 shows a list of the twenty-five MLAs used in the lazypredict library.

The lazypredict library contains MLAs falling into the following categories (Bali et al., 2018):

- Linear models such as logistic regression (LR), Naïve Bayes (NB), and support vector machine (SVM).
- Non-parametric models such as k-nearest neighbor (kNN).
- Artificial Neural Networks (ANN) (such as multilayer perceptron).
- Tree based methods such as decision tree (DT).
- Ensemble bagging methods such as random forest (RF).
- Ensemble boosting methods such as gradient boosting machines including XGBoost (XGB), LightGBM (LGBM) and AdaBoost (AB).

AdaBoostClassifier
BaggingClassifier
BernoulliNB
CalibratedClassifierCV
DecisionTreeClassifier
DummyClassifier
ExtraTreeClassifier
ExtraTreesClassifier
GaussianNB
KNeighborsClassifier
LGBMClassifier
LinearDiscriminantAnalysis
LinearSVC
LogisticRegression
NearestCentroid
NuSVC
PassiveAggressiveClassifier
Perceptron
QuadraticDiscriminantAnalysis
RandomForestClassifier
RidgeClassifier
RidgeClassifierCV
SGDClassifier
SVC
XGBClassifier

Table 2.1: Machine Learning Models in the lazypredict package.

Using the benchmarks obtained from training the twenty-five MLAs as a guide, several baseline models were considered as potential models for further improvements through the use of cross validation, hyperparameter tuning and ensemble learning techniques which are discussed in the sections to follow. The benchmarks were based on four metrics: accuracy, balanced accuracy, F1 score and ROC AUC score. These metrics are discussed in detail in Chapter 3, section 3.4.4.

2.4.2 Cross Validation

The basic premise behind ML modelling is to develop a generalized model on existing data which then performs well in predictions on unseen data (Bali et al., 2018). One of the most common strategies used to simulate unseen data is k-fold cross validation.

The steps for the k-fold cross validation technique are as follows (Kakarla et al., 2021):

1. Split the entire dataset randomly into k-folds
2. Use (k-1) folds collectively as the training set and the kth left out fold as the test set
3. Apply performance metrics such as accuracy on the model
4. Repeat the process until all of the folds are used as a test dataset
5. The final metric is calculated as the average of all the k-fold metrics

The process is depicted in Figure 2.3 using 10 folds as an example and some metric *E*.

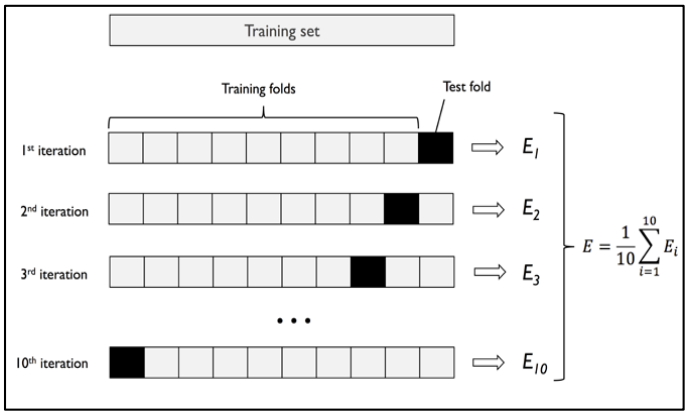

Figure 2.3: 10-fold cross validation. Source: Raschka and Mirjalili (2017)

The 10-fold cross validation strategy was adopted for this study.

2.4.3 Hyperparameter Optimization

Hyperparameters are meta parameters which are set before model training and building ML models (Bali et al., 2018; Zheng, 2015). In order to improve the performance of MLAs, the hyperparameters can be optimized or tuned. Hyperparameters are not dependent on the dataset used for training a ML model.

Two common type of methods to search the parameter space for optimal hyperparameters are Grid Search and Random Search. In a Grid Search, a grid of hyperparameter values are chosen and an exhaustive search is performed on the entire grid of values to return the best hyperparameters (Zheng, 2015). This can be computationally expensive, especially when the search space is large (Géron, 2019). Rather than search the entire grid, Random Search, which is a variation of the Grid Search, evaluates a random sample of points on the grid and is not as computationally expensive as a Grid Search. Despite not searching the entire grid, a study conducted by Bergstra and Bengio (2012) found that Random Search was as effective as Grid Search in a large number of cases.

The Random Search hyperparameter optimization method was adopted for this study.

2.4.4 Ensemble Learning

While every classifier has its strengths and weaknesses, the goal of ensemble learning is to combine different classifiers into a stronger meta-classifier or ensemble in order to solve a shared problem (Raschka & Mirjalili, 2017). Aggregating the predictions of a group of classifiers is expected to provide a better generalization performance than each of the individual classifiers on their own (Beyeler, 2017; Géron, 2019; Raschka & Mirjalili, 2017). This is based on the premise that instead of relying on a single expert (classifier), predictions of individual experts (classifiers) can be strategically combined to come up with a more accurate prediction (Polikar, 2012). In order to control the classification error, there are two parts that can be controlled, namely, the bias which is the classifier's accuracy, and the variance which is the classifier's precision when trained using different datasets. Classifiers with low bias usually have a high variance and vice versa. Thus the idea behind ensemble techniques is to create several classifiers with similar bias and averaging their outputs to reduce the variance (Polikar, 2012).

An ensemble will usually contain two major components: a set of classifiers and a set of decision rules to determine how the results of the classifiers should be combined to give a

single output. One common ensemble method is the averaging method where models are developed in parallel and a combined estimator is determined through averaging or voting schemes (Beyeler, 2017, p. 264).

A voting classifier ensemble method was adopted for this study. Figure 2.4 depicts a workflow of the voting ensemble. Two voting schemes were considered, namely hard voting and soft voting.

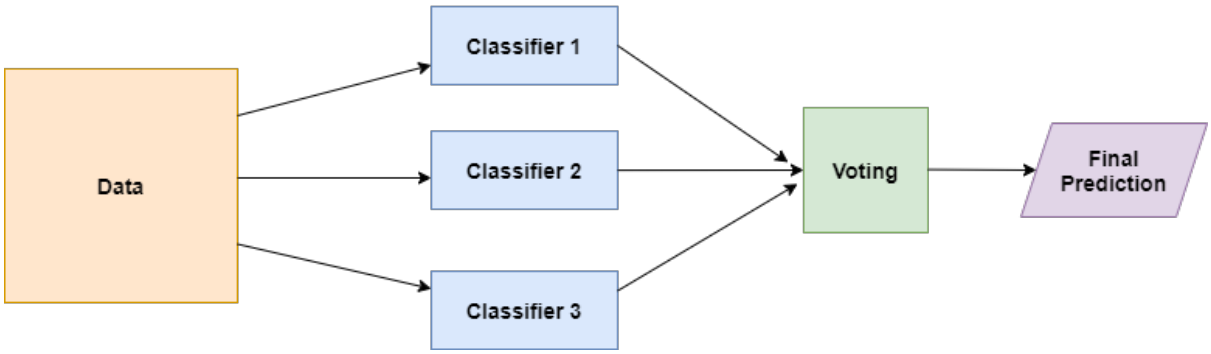

Figure 2.4: Workflow of a voting classifier. Source: Researcher's own creation.

2.4.4.1 Hard Voting

In hard voting, (or majority voting), each classifier votes for a class (such as 0 or 1, in the case of binary classification), and winner is the majority vote or the mode or the most frequently occurring of the predicted class labels (Beyeler, 2017, p. 292). Thus if there were 3 classifiers, classifier 1, classifier 2 and classifier 3 predicting 1,1 and 0 respectively, then the mode is 1. The ensemble classifier will predict class 1.

2.4.4.2 Soft Voting

In soft voting, each classifier calculates a weighted probability value predicting that an instance of a dataset belongs to a specific target class (such as 0 or 1, in the case of binary classification). The predictions are weighted by the classifier's importance and all weighted predictions are summed up (Beyeler, 2017, p. 292). The target class label (0 or 1) is then determined to be the one with the greatest sum of weighted probabilities. As an example of how prediction works for a binary class with labels 0 or 1, consider three classifiers with the ensemble making the following predictions as shown in Table 2.2

Classifier	Class 0	Class 1
Classifier 1	0.9	0.1
Classifier 2	0.7	0.3
Classifier 3	0.4	0.6

Table 2.2: Soft voting prediction example. Source: Researcher's own creation.

If the weights {0.1, 0.1, 0.8} are respectively assigned to each classifier, the weighted average can be calculated for each class as follows:

- For class 0, the weighted average is 0.1*0.9+0.1*0.7+0.8*0.4 = 0.48
- For class 1, the weighted average is 0.1*0.1+0.1*0.3+0.8*0.6 = 0.52

Since the weighted average for class 1 is 0.52 which is higher than the weighted average for class 0 which is 0.48, the soft voting ensemble classifier will predict class 1.

Since soft voting gives more weight to highly confident votes, it generally achieves higher predictive performance than hard voting (Géron, 2019).

The soft voting ensemble scheme was chosen for this study.

The next section evaluates some recent approaches used to identify SAR using ML. These publications were chosen based on the varying ML approaches and MLAs used in attempts to identify SAR.

2.4.5 A Summary of Machine Learning Approaches Applied to Identifying Students at Risk

The following review of existing literature provides some guidance as to the ML approaches and algorithms employed to identify SAR. A tabular summary is provided in Table 2.3.

Lakkaraju et al. (2015) used RF, AB, LR, SVM, and DT to identify SAR in two USA school districts comprising over 20 000 students. The authors used multiple student attributes that were fed to the MLAs to identify SAR.

Agrawal and Mavani (2015) analysed the experiments of three other researchers and found that the Naive Bayes (NB) worked well at predicting student performance. The authors argued that

ANN also performed just as well. They noted that the performance of the ANN improved with a greater training dataset size.

Educational Data Mining (EDM) was employed by Acharya and Sinha (2014) in the early prediction of students' results. EDM involves the application of ML, data mining and statistics to data created for educational environments such as universities. The authors first identified a set of attributes from students majoring in computer science in some of the undergraduate colleges in Kolkata, India. Feature Selection was then used to reduce the attribute set of features. Of the different classes of MLAs used, such as DT, NB, ANN, SVM, it was found that the DT algorithm performed the best with a 66% F1 score.

Chen, Hsieh, and Do (2014) used ANN with 2 meta-heuristic algorithms inspired by cuckoo birds and their behaviour, namely Cuckoo Search (CS) and Cuckoo Optimization Algorithm (COA), to predict student academic performance. The authors found that the use of the COA together with ANN showed a slightly improved performance over the use of just the ANN to predict student academic performance. Eight class predictor variables were used and it was suggested that prediction accuracy could possibly be improved with the addition of other additional attributes, such as character, intelligence, and psychological factors.

Rovira et al. (2017) used data from degree studies of students in law, computer science and mathematics at the Universitat de Barcelona in Spain, to predict student dropout rates The authors used LR, NB, SVM, RF, and AB. The highest prediction accuracies achieved were 82% for law, 76% and 61% for computer science, and 61% for mathematics. Since the sample sizes for the computer science and mathematics degrees were relatively small (416 and 516 respectively), the authors suggested a non-parametric model such a NB or LR for training

In one of the earlier works on the use of ML for dropout prediction, Kotsiantis, Pierrakeas, and Pintelas (2003), used six MLAs, namely DT, ANN, NB, LR, SVM, and Instance-Based Learning (IBL) to predict student dropout rates in distance learning systems. At different stages, the number of attributes used in the models was increased gradually, and comparisons were made to find the best fit. The authors concluded that the prediction accuracy, which was 63% using only demographic data, improved by the middle of the semester, when additional attributes were added to the training set. The NB algorithm with an accuracy of 83% proved to be the most appropriate algorithm.

Lykourentzou et al. (2009) used a combination of three MLAs, namely Feed-Forward Neural Network (FFNN), SVM, and Probabilistic Ensemble Simplified Fuzzy ARTMAP. Three different decision schemes were used to combine the results of the ML techniques to investigate if it would improve classification performance. The authors concluded that this approach increased the classification accuracy with a claim in excess of 97% accuracy.

Er (2012) built on the studies of Kotsiantis et al. (2003) and Lykourentzou et al. (2009) to propose a model to identify SAR using ML in a course called IS 100. The algorithms used were IBL, DT and NB. The author also additionally used three decision schemes to combine the results of the ML techniques in different ways, to investigate if better classification could be achieved. It was concluded that combining the results of the algorithms using decision schemes may produce better prediction results, than just using a single algorithm at a time.

Livieris, Drakopoulou, and Pintelas (2012) used FFNN to predict the performance of students on a mathematics course in a Greek institution. The generalization accuracy of the FFNN was compared with other classifiers such as DT, Bayesian Networks, Classification Rules, and SVM. The input variables were restricted to just the student results obtained during an entire year. The authors claimed to have achieved better results, using the FFNN, than the other classifiers.

Koutina and Kermanidis (2011) used six different classification algorithms namely DT, KNN, NB, RIPPER, RF, and SVM on five academic courses to predict the final grade of Ionian University informatics postgraduate students. A combination of demographic data and term results were used as training data. Due to small sample sizes and class imbalances, the data had to be resampled. The prediction accuracy was 85%.

Oladokun, Adebanjo, and Charles-Owaba (2008) used ANN to predict student performance at the University of Ibadan in Nigeria using ten attributes which were made up of demographic data, as well as previous results. The network had a 74% prediction accuracy. The authors stated that one of the limitations of the model was the lack of relevant performance-influencing factors, which could not be obtained from the pre-admission forms they used to extract the other data.

Cheewaprakobkit (2015) used DT and ANN to predict the academic performance of 1600 students from a university in Thailand. Twenty-two student attributes were used. The results showed that DT had a slightly higher accuracy of around 85%, compared to ANN, which had an accuracy of around 84%.

Al-Obeidat, Tubaishat, Dillon, and Shah (2017) used a hybrid classification technique that combined DT and fuzzy multi-criteria classification to predict student performance using previous results and demographic data, such as age, school, address, and family size. The classifier was compared to other classifiers such as RF, NB, Meta Bagging, Attribute Selected Classifier, Simple Logistic and Decision Table. The authors claimed the algorithm had a higher accuracy than all the other classifiers, with an accuracy of 82.28%.

Bayer, Bydzovská, Géryk, Obsivac, and Popelinsky (2012) attempted to use students' social behaviour with ML techniques to improve the prediction of dropouts and failure rate of 775 students at the Masaryk University in the Czech Republic. The accuracy rate for dropout prediction was not more than 69% using just social behaviour data. This increased to closer to 80% when student data was added.

Asif, Merceron, and Pathan (2014) used DT, NB, and ANN to predict the academic performance of 347 undergraduate students in their fourth year of study at a Pakistani university. The study was limited to using just marks while excluding socio-economic data for the prediction. This resulted in an accuracy rate of 60.5%

Siri (2015) used ANN to predict the dropout rate of 552 health care students at the University of Genoa in Italy. The results showed a 76% prediction accuracy for dropout prediction.

Chai and Gibson (2015) used DT, LR and RF on 23291 students at the Curtin University in Australia to identify SAR. The best prediction was from RF with 67% accuracy. The scope of the analysis was for the first semester of students' first year. The authors claim an improvement in the prediction accuracy at the end of the semester, since there was more student data available to analyse.

Berens et al. (2018) used ML methods to develop an early detection system to identify SAR on students at the Federal State of North Rhine-Westphalia university in Germany using administrative student data. An ensemble of AB, ANN and DT was used. Prediction accuracy

in semester one (S1) was 79%. This improved to 90% by the end of semester four (S4). The authors attributed this improvement to the availability of more data to analyse.

Bayer et al. (2012) used students' demographic and social behaviour from the Information System of Masaryk University in the Czech Republic, to predict SAR over seven semesters using the ZeroR, NB, SMO, IB1, OneR, PART, and J48 algorithms. Some algorithms performed better than the others depending on the semester. The highest prediction accuracies ranged from 72.40% in S1 to 93.51% in semester seven (S7). This followed a similar trend to the study by Berens et al. (2018), where the addition of more time over the different semesters seemed to have improved prediction accuracies.

Kabathova and Drlík (2021) employed six different MLAs, namely LR, DT, NB, RF, ANN, and SVM to predict student dropout on 261 students at the Constantine the Philosopher University in Slovakia. The authors used VLE data from the Moodle platform which consisted of course views, test, and assignment scores. The lowest prediction accuracy was 77% using NB and 93% using RF.

A review of the literature reaffirms the position that no one specific MLA performs best in every circumstance. Therefore, literature alone may not be sufficient when deciding on the choice of algorithms for ML. Several other factors can impact on the performance of MLAs. Some of these factors are discussed next.

2.4.6 Factors Affecting the Performance of Machine Learning Algorithms

The literature shows growing academic interest over the years, yet, following similar methodologies, when it comes to the use of ML techniques to identify SAR. This is showcased in Table 2.3. Berens et al. (2018) stresses that the studies are not always comparable, owing to various reasons, such as different sample sizes, the settings applied to the variables, and even the general research methods and questions used. It was noted by the authors that despite these reasons, the studies often show only minor differences in prediction accuracy. The authors noted that the bigger differences between the studies are from the data itself, rather than the method of prediction. Ideally, as much demographic and social data, combined with academic data of students, need to be collected.

Some authors used the integration model of Tinto (1975) to collect the relevant data that best described the social and academic integration of a student at a HEI. This is not always possible with every student, and as a result, not all of the possibly relevant data may always be available to be used in any prediction model. Furthermore, while some information may be crucial to help improve the prediction accuracy of identifying SAR, data privacy and protection laws may prevent the use of such data. This is a limitation of any study, and it needs to be investigated via our models, how much of an impact these limitations will have on the ability of the model to accurately predict SAR.

ML has at its epicentre, algorithms. The performance of a specific algorithm can be affected by several factors, such as the size of the dataset (Acharya & Sinha, 2014; Agrawal & Mavani, 2015; Al-Obeidat et al., 2017; Koutina & Kermanidis, 2011; Oladokun et al., 2008), and class imbalances (Koutina & Kermanidis, 2011; Soobramoney & Singh, 2019; Sun, Wong, & Kamel, 2009), where there is a high variance in the class sizes and the number of features, as well as the correlation between the different features of the datasets. If there are too few of what may be relevant features, this may result in the poorer performance of an algorithm (Oladokun et al., 2008). Highly correlated features can also affect the training of the algorithms, as it will introduce noise into the system, resulting in over-fitting of the data. This will also result in poorer performance when the model is tested on unseen data (Lee & Chen, 2018; Lukman, Ayinde, & Ajiboye, 2017; Nilashi, Ibrahim, Ahmadi, Shahmoradi, & Farahmand, 2018; Ouyang et al., 2019; Shahbaz, Rasi, Ahmad, & Management, 2019; Solares, Wei, & Billings, 2019; Xu, Esquerre, & Sun, 2018). A large dataset is preferable as there is more data to train the algorithms on (Chicco, 2017; Cichos, Gustavsson, Mehlig, & Volpe, 2020; Nilashi et al., 2018). However, if the dataset has class imbalances, where one output class type is much greater than the other, then the algorithm would be biased towards that specific class during training (Ali et al., 2019; Li et al., 2019; Priya & Uthra, 2021). There are several factors that will need to be considered in order to get successful performance from MLAs, for the prediction of students at risk. In this regard, measuring students at risk becomes a non-trivial task.

2.4.7 Related Works and Gap in the Literature

Several studies used demographic and VLE data from a dataset called OULAD to identify SAR. Hlosta et al. (2017) used NB, LR, SVM, RF and XGB MLAs for the early identification of SAR. XBG performed best with an F1 score of 71%. Hussain, Zhu, Zhang, and Abidi

(2018b) attempted to identify SAR by analysing students' engagement with the VLE. Six MLAs namely DT, JRip, J48, CART, NB and gradient-boosted trees (GBT) were used with J48 having the highest accuracy of 88.52%. The studies by Haiyang et al. (2018) and Heuer and Breiter (2018) focused on identification of SAR through students' daily VLE interactions. Prediction accuracy was 90.85% using DT, RF, LR and SVM MLAs. Azizah et al. (2018) compared the performance of C4.5 and NB MLAs to predict students' performance. NB performed slightly better of the two, with an accuracy of 63.8%. Wasif et al. (2019) used SVM, LR, NB and RF MLAs to predict the performance of students based on their logging history with the VLE. With a prediction accuracy of 89%, RF performed the best. The study by Rizvi et al. (2019) investigated the role that demographics played in online learning using the DT MLA. A prediction accuracy of 83.14% was achieved. Aljohani et al. (2019) investigated the use of deep-learning techniques to predict SAR in a VLE. This was done by employing a long short-term memory (LSTM) recurrent ANN. F1 scores varied from 60.16% in the first week to 83.70% in the last week of the course. Chui et al. (2020) used a reduced training vector SVM to predict SAR. Unnecessary training vectors were removed in order to speed up training time. Overall accuracy of 91.3% was achieved.

After having conducted a review of some currently adopted MLA-based studies that used OULAD to identify SAR in a VLE, it was found that while these solutions have value, they are not without limitations. The studies either did not use cross validation or used a maximum of 5-fold cross validation. None of the studies performed any hyperparameter optimization. Several of the studies also used very small sample sizes. The accuracy scores for all of the studies never exceeded 92%. Only one study by Aljohani et al. (2019) used weekly VLE data for predictions. This was also the only study that used LSTM recurrent ANN as opposed to conventional classifiers. None of studies reviewed used any ensemble learning techniques. A summary of these studies is given in Table 2.3, together with some of their limitations and problems.

The limitations of these studies lead to the development of a MLA-based SAR prediction model using an ensemble of best performing MLAs for early and accurate prediction of SAR.

Table 2.3: Comparison of the prediction accuracy of various studies using OULAD. Source: Researcher's own creation.

Author (S) and Year of Publication (Ascending)	**Aim of Study**	**Limitations, Problems or Challenges with Study**	**Sample Size**	**Utilized MLAs**	**Prediction Accuracy**
Hlosta et al. (2017)	Early identification of SAR	Small sample size (7% of total); single dataset used; low prediction score; Conventional MLAS used as final classifiers; No hyperparameter tuning	2500	NB, LR, SVM, RF, XGB	XGB 71% F1 score
Hussain et al. (2018b)	SAR in a VLE	Very small sample size (1% of total); Single dataset; Conventional MLAS used as final classifiers; No demographic data used, only VLE; No hyperparameter tuning	383	DT, JRip, J48, CART, NB, GBT	J48 88.52%
Haiyang et al. (2018)	SAR through daily VLE interactions	No pre-processing since full dataset with anomalies specified as sample size; Only DT classifier used; No demographic data used only VLE; No hyperparameter tuning of models	32593	DT	DT 90%
Heuer and Breiter (2018)	SAR through daily VLE interactions	No hyperparameter tuning; 5-fold cross validation	32593	DT, RF, LR and SVM	SVM 90.85%
Azizah et al. (2018)	Student performance	Small sample size; Only two conventional MLAs; No hyperparameter tuning; Low prediction accuracy	1700	C4.5, NB	NB 63.5%

Wasif et al. (2019)	Student performance based on logging history	No pre-processing since full dataset with anomalies specified as sample size; No hyperparameter tuning; Single dataset used; No demographic data used, only VLE	32593	SVM, LR, NB, RF	RF 89%
Rizvi et al. (2019)	Role of demographics in students' performance	Small data set; Study focused on demographic and not VLE data; No hyperparameter tuning; only DT MLA used	8581	DT	DT 83.14%
Aljohani et al. (2019)	SAR using clickstream data in a VLE	No demographic data used, only VLE; Relatively low F1 scores; No hyperparameter tuning	22437	LSTM	F1 score 60.16% (W1) F1 score 83.7% (W38)
Chui et al. (2020)	SAR using data in a VLE	No hyperparameter tuning, 5-Fold cross validation on selected data, only SVM MLA used, Single dataset used	32593	SVM	SVM 92%

2.5 Summary

This chapter provided a review of the literature with regards to SAR from a local, as well as a global perspective. The impact of SAR was analysed from different perspectives: the students, HEIs, and the RSA economy. The need for pursuing the problem of identifying SAR was explained. Some of the current manual or statistical approaches of attempting to identify SAR were highlighted. A comparative performance analysis of MLAs used in predicting SAR which worked towards achieving RO1. The shortcomings or challenges of the current manual or statistical approaches were explored and presented. MLAs and their use were explained in detail. A review of the literature on the use of ML techniques to identify SAR was done. Some of the limitations from this review were used to propose a more robust MLA-based SAR prediction model using an ensemble of best-performing MLAs, as called for in RO2. The next chapter will explain the research methodology that was followed in order to achieve the research objectives of this study.

Chapter 3: A Description of the Research Framework

3.1 Introduction

This chapter presents the components of the research framework adopted for this study. In this chapter, the reader is presented with the justification for positioning this research in the post-positivism paradigm (3.2), and the motivation for situating it as quantitative research (3.3). This is followed by a detailed description of the methodology commissioned for the development of a predictive ML model for the early identification of SAR. The choice of metrics and ML classification techniques used to evaluate the effectiveness of the ML model (3.4) follows. Then the credibility of the research is discussed in terms of validity and reliability (3.5). The chapter ends with a summary of the research framework (3.6). Table 3.1 provides a broad overview of the components of the research framework.

Table 3.1: Overview of the research framework chosen for this study.

Component	**Description**
Paradigm	Post-positivism
Research strategy	Quantitative

3.2 Positioning this Study in the Post-positivism Research Paradigm

There are various philosophical beliefs or ways of thinking that guide the research process, from choosing a topic to research, to the publication of the final research findings. These beliefs combine to form the fundamental basis of research. A paradigm can be understood as a set of beliefs that represent "a worldview or framework through which knowledge is filtered" (Leavy, 2017, p. 11). There are different research paradigms. Creswell and Creswell (2018) juxtapose positivism, post-positivism, constructivism, transformative, and pragmatism. Research endeavours are often *anchored* in one of these paradigms. Two of these paradigms, namely positivism and post-positivism, were of interest for this study. The following sub-sections provide a discussion of each candidate paradigm. Thereafter, justification is provided for placing this study in the post-positivist paradigm.

3.2.1 An overview of Post-positivism

Natural science research in the past was mostly anchored in the positivism paradigm. Leedy and Ormrod (2019) argue that positivism is based on the belief that with the correct type of

tools (in a deterministic system), where there is an underlying cause for every effect, the truth can be objectively uncovered. Positivism is therefore built on the belief that everything is based on objective facts and human biases do not exist. This view, however, seems to be at odds with human nature, where a researcher may unintentionally introduce certain biases, for instance, when it comes to the best ways to measure variables, or the type of inferences one can make from the data.

Due to the perceived shortcomings of positivism, some researchers instead subscribe to the post-positivism paradigm, which is derived from positivism. Post-positivism involves reducing ideas and quantitative data into small, discrete structures that can be precisely measured, for instance, through experiments. Creswell and Creswell (2018) argue that when it comes to human beings and their behaviour, one cannot claim absolute certainty about knowledge. Phillips and Burbules (2000) further challenges the idea that there could be a claim of absolute truth when it comes to knowledge. Thus, Creswell and Creswell (2018) point out that while post-positivists may also subscribe to a deterministic philosophy, where there is a need to identify the causes of certain outcomes, like those found in experiments, Leedy and Ormrod (2019) suggest that post-positivists do not lay claim to absolute truth of having proven something. They are more likely to suggest that their experimental findings increase the probability of something being true.

3.2.2 Justification for Situating this Study in the Post-positivism Paradigm

Creswell and Creswell (2018) posit that post-positivists follow the scientific method towards research, whereby the researcher starts with some theory about the research, then collects data that will either support or refute that theory, and then makes any revisions before further tests take place. This study aimed to use ML techniques to identify SAR. The literature review provided initial information about the possible different characteristics that may be used in order to identify SAR. The appropriate dataset was obtained and prepared. Experiments were then performed on the dataset using ML techniques to obtain results that were evaluated. Revisions and further tests on the data were performed as part of the ML process. This study is thus placed in the post-positivism paradigm.

3.3 A Quantitative Research Strategy for this Study

A research strategy is a systematic plan that a researcher follows to conduct research (Easterby-Smith, Thorpe, & Jackson, 2012; Johannesson & Perjons, 2014). It refers to the manner in which data is collected, analysed, and reported on (Mackenzie & Knipe, 2006). There are three

common research strategies: qualitative, quantitative, and mixed methods. The focus of discussion in the next section will be on the quantitative strategy, with a view to inform the choice of research strategy for this study.

3.3.1 Quantitative Strategy

A quantitative research strategy entails testing theories by looking at the relationship between different variables. If these variables contain numeric data, they can be measured and statistically analysed using various instruments (Bryman, 2012; Goertzen, 2017). The results should be precise, reliable, and generalizable, with less bias (Creswell & Creswell, 2018). Quantitative data analysis involves four main kinds of data, namely nominal, ordinal, interval, and ratio. Nominal (or categorical) data describes different categories with no numerical ranking (such as 0 for male and 1 for female). Ordinal data has a quantitative ranking scale (such as 0 for no qualification, 1 for undergraduate, 2 for postgraduate). Interval data also has a quantitative scale, but with the difference between the points of scale having consistently the same size. Ratio data is similar to interval data, except that it has a true zero on the measurement scale (Oates, 2006). This strategy works with the post-positive paradigm, since one is dealing with a deterministic system (Leedy & Ormrod, 2019; Oates, 2006). If a researcher is trying to verify what impact a set of independent variables (cause) has on a dependent variable (effect), a quantitative strategy is preferable (Rubin & Babbie, 2016). With the quantitative strategy, a researcher is able to generalize, replicate, and retest findings, since the analysis of the data is based on actual quantities that can be measured and verified by other researchers using statistical tests (Creswell & Creswell, 2018; Kumar, 2011; Oates, 2006).

3.3.2 Justification for The Use of the Quantitative Strategy

The task of identifying SAR involves training ML models, by providing MLAs with training data as input, in order to learn from the data and predict an output which shows a student being at risk or not at risk. The training data for this study comprises students' demographic and VLE interactions data. All input data must be numeric before being used as training data. Thus, non-numeric demographic data, such as gender and qualification, is converted to numeric values.

The output can be measured and analysed through the use of various ML metrics. These metrics are determined from mathematical equations and give objective outputs based on the calculations. It is thus evident from the process of data preparation, to the use of MLAs to train

the data in order to predict an output and evaluating the results, that the quantitative research strategy is applied. The quantitative research strategy is thus adopted for this study.

3.4 Setting up the Machine Learning Experiment

The task of identifying SAR using ML techniques is a binary classification, supervised ML task, whereby a student is determined to either be at risk or not at risk. It involves training the ML models on a set of data, and then using the trained models on new data to make predictions. Figure 31. illustrates the base supervised ML process used in this study

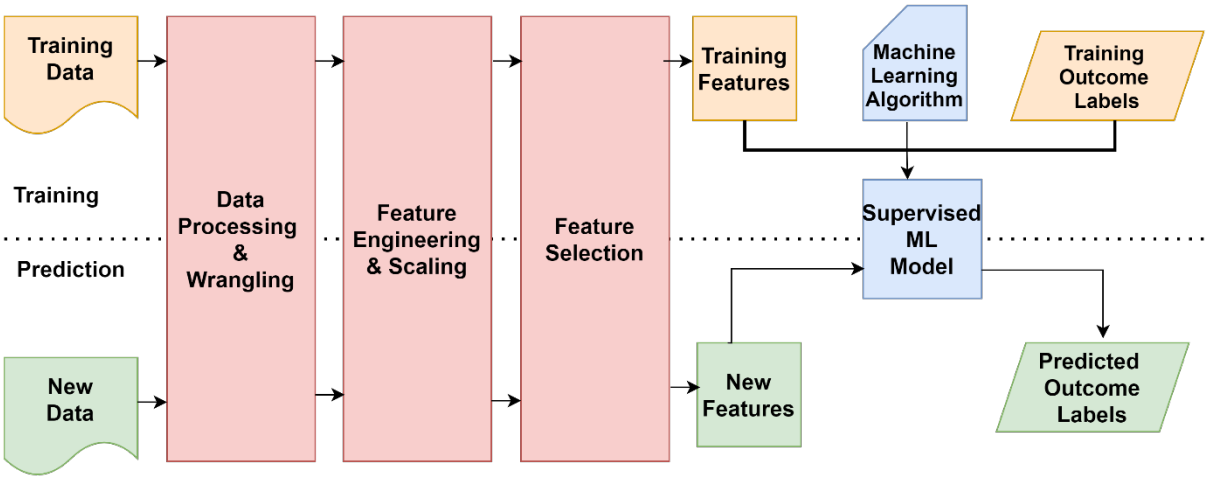

Figure 3.1: Supervised machine learning pipeline. Source: Adapted from Bali et al. (2018).

The next sub-sections describe the different aspects of the experiment that was set up to determine the suitability of ML techniques to identify SAR. A description of the chosen dataset, data processing, feature engineering and scaling, feature selection, and metrics that will be used for this study, is provided.

3.4.1 Data Wrangling and Processing

In order for ML models to make accurate predictions, the raw student data requires pre-processing. Data wrangling which involves gathering, selecting, cleaning, and transforming data as part of the ML process, will be discussed in the next sections 3.4.1.1 to 3.4.1.3.

3.4.1.1 Data Acquisition and Description

Due to privacy and ethical reasons, actual student data is not readily available online. An extensive review of the literature and other sources yielded no relevant datasets that could be used to identify SAR in the RSA HEI context. In addition, despite the researcher's best efforts, no studies were found to investigate SAR from the VLE perspective and therefore those

datasets were deemed not entirely feasible for this study. The literature review has also shown that SAR is a problem not only in RSA HEIs, but in most HEIs all over the world. The factors that could play a role in identifying SAR are common to students in different HEIs worldwide, with very subtle differences (mainly in the terminology). In this regard, the dataset that was found to be suitable for this study was obtained from an online university in the UK. It is called the Open University Learning Analytics Dataset (OULAD)[1], which contains student demographic and interaction data with the university's VLE (Kuzilek, Hlosta, & Zdrahal, 2017).

OULAD contains data about students, the courses they take, and their interactions with a VLE for seven selected courses (called modules) for the 2013-2014 academic year. The original dataset consists of 32 953 registered students, with 22 courses and 43 variables. It is a subset of the student data contained in the databases of the university, and was extracted specifically for research purposes. The dataset that was extracted does not contain any personally identifiable student information, as it went through an extensive data anonymization process before being released for research use. Thus, there are no ethics clearance issues.

Students could take different modules and the typical duration of a module was approximately 36 weeks. However, since students were allowed to access the VLE resources a few weeks before a module could start, the duration was closer to 40 weeks. Table 3.2 contains a summary of the dataset.

Table 3.2: Summary of the OULAD. Source: Adapted from Kuzilek et al. (2017).

CSV Data File	**No. of Instances (Rows)**	**Description of Data File**	**Attributes**
studentInfo	32 593	Student demographic information and final result for a course that was taken	code_module, code_presentation, id_student, gender, region, highest_education, imd_band, age_band, num_of_prev_attempts, studied_credits, disability, final_result
courses	22	The courses taken, start dates and duration in days.	code_module, code_presentation, length
studentRegistration	32 593	Courses a student is registered for and their registration date	code_module, code_presentation, id_student, date_registration, date_unregistration
assessments	206	Assessment types and weights	code_module,code_presentation, id_assessment, assessment_type, date, weight

[1] https://analyse.kmi.open.ac.uk/open_dataset

studentAssessments	173 912	Assessment scores from students	id_assessment,id_student, date_submitted, is_banked, score
studentVle	10 655 280	Every student interaction (called sum_clicks) with the VLE resources, including date of interaction	code_module,code_presentation, id_student, id_site, date, sum_click
vle	6364	The different type of VLE resources available	id_site,code_module, code_presentation, activity_type, week_from, week_to

The attributes shown in Table 3.2 reflect the type of information stored in each of the seven different data files. The dataset was deemed suitable for this study, since it contains the necessary demographic and VLE interaction data required for use in the ML process of this study, to be able to identify SAR. The attributes in the dataset are also standard attributes that may be found in any dataset of a similar structure, irrespective of the country of origin. Some of the attribute names may have different terminology from, for example, a similar dataset in the SA context. The attribute called imd_band for instance is a case in point. It refers to the English Indices of Deprivation measure which measures deprivation levels in 32 844 different neighbourhoods in the UK (Bowie, 2019). The deprivation levels are ranked according to poverty levels and stored in the imd_band variable. In the RSA context, different areas of the country could be ranked in a similar fashion. Simply put, each attribute in the dataset can be mapped to equivalent attributes in the RSA context.

The acquired data will have to be processed before it becomes suitable for use in ML to identify SAR. The next section discusses the data processing steps.

3.4.1.2 Data Cleaning

Data cleaning involves filtering out irrelevant data, such as missing records, duplicated records, and incorrectly formatted data. Inaccurate data that is input to the ML models can result in inaccurate predictions and decision making. In order for the predictive model to be robust, data cleaning is thus a crucial part of the process. Jesmeen et al. (2018) pointed out that failure to clean data before using it can lead to inaccurate and unpredictable results.

To perform optimally, MLAs, in general, require a dataset to contain only numeric values. All non-numeric data will require conversion to numeric data.

3.4.1.3 Exploratory Data Analysis

The exploratory data analysis (EDA) step helps a researcher to understand and gain insights into the dataset through visualization or statistics. This allows a researcher to discover any patterns, or even anomalies, in the data. Univariate plots (such as box and whisper or histograms) can be used to gain insights on individual variables, while multivariate plots (such as scatterplots) can be used to understand the relationships between attributes.

3.4.2 Feature Extraction, Engineering, Scaling, and Selection

Extracting important features or attributes, or creating new features from the existing features, can help to create a better ML model. If the dataset contains high numeric variation, to prevent bias towards the high values during the ML training process, some of the data attributes need to be normalized and scaled within a certain range, which is usually a range between 0 and 1 Bramer (2013). A commonly used formula for scaling is the *min-max* scaling which is shown in Equation 3.1

$$x_{norm} \frac{x - x_{min}}{x_{max} - x_{min}} \quad 3.1$$

Feature selection needs to be performed to obtain a subset of the important features.

3.4.3 Machine Learning Modelling and Classification

In order to evaluate the predictive capabilities of the MLAs, twenty-five different MLAs will be used through a Python package called Lazy Predict (Pandala, 2020), which fits and evaluates all ML models from Python's scikit-learn package.

3.4.4 Evaluating Machine Learning Model Performance

Evaluating the performance of a ML model can be done using several classification metrics (such as accuracy, precision, recall, F1 score, and ROC AUC score). These values can be derived from a confusion matrix.

3.4.4.1 Confusion Matrix

A confusion matrix summarizes the performance of a ML model in a tabular form in an NxN matrix. For a binary classification problem with 2 classes, such as students at risk and students not at risk, the confusion matrix is a 2x2 matrix. The confusion matrix for identifying SAR is shown in Table 3.3.

Table 3.3 Confusion matrix for identifying SAR. Source: Researcher's own creation

		Actual	
		At Risk	**Not At Risk**
Predicted	**At Risk**	True Positive (TP)	False Positive (FP)
	Not At Risk	False Negative (FN)	True Negative (TN)

The actual values are depicted on the columns and the predicted values are depicted on the rows as follows:

- True positive (**TP**) counts the number of SAR correctly classified as being at risk
- False Positive (**FP**) counts the number of SAR misclassified as being at risk
- True negative (**TN**) counts the number of students not at risk, correctly classified as being not at risk
- False Negative (**FN**) counts the number of SAR misclassified as being not at risk

3.4.4.2 Accuracy

Accuracy is a performance metric that can be used to measure the performance or predictive capabilities of a ML classifier. The accuracy scores, however, are reliable when the dataset has balanced or nearly balanced classes. Accuracy refers to the ratio of correct predictions to the total number of predictions. In other words, of all the predictions that were made for identifying SAR, accuracy asks the question: "What ratio of students were correctly identified as either being at risk or being not at risk?" Equation 3.2 shows the formula to calculate accuracy:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad 3.2$$

Accuracy as performance measure becomes less useful when there are highly imbalanced classes, and other performance metrics need to be used for more meaningful evaluation of the results (Pes, 2020; Picek, Heuser, Jovic, Bhasin, & Regazzoni, 2018). Some of these metrics are discussed in the sections to follow.

3.4.4.3 Balanced Accuracy

Balanced accuracy is another performance metric that can be used to measure the performance of a ML classifier. It is more useful than accuracy when there is a class imbalance. Balanced accuracy is the arithmetic mean of two other metrics called sensitivity (also known as the true positive rate or recall) and specificity (also known as the true negative rate). Equations 3.3, 3.4 and 3.5 show the formulas to calculate sensitivity, specificity, and balanced accuracy respectively.

$$Sensitivity\ (Recall) = \frac{TP}{TP + FN} \tag{3.3}$$

$$Specificity = \frac{TN}{FP + TN} \tag{3.4}$$

$$Balanced\ Accuracy = \frac{Sensitivity + Specificity}{2} \tag{3.5}$$

Sensitivity asks the question "Out of all the students that were actually at risk, what ratio of students were correctly identified as being at risk?". Specificity asks the question: "Out of all the students that were actually not at risk, what ratio of students were correctly identified as being not at risk?"

3.4.4.4 F1 Score

The F1 score is the harmonic mean or weighted average of two other metrics, called precision and recall. Recall (or sensitivity) was discussed in section 3.4.4.3. Precision is the ratio of students correctly predicted as being at risk, to the total number of students predicted as being at risk. In other words, precision asks the question: "Of all the students predicted as being at risk, what ratio of students were correctly predicted as being at risk?" Equation 3.6 shows the formula to calculate precision, while Equation 3.7 shows the formula to calculate the F1 score:

$$Precision = \frac{TP}{TP + FP} \tag{3.6}$$

$$F1\ Score = 2 * \frac{(Recall * Precision)}{Recall + Precision} \tag{3.7}$$

The F1 score is also more useful than accuracy, when there is a class imbalance, since it takes into account both the false positives as well as the false negatives. This provides for a balanced optimization between precision and recall.

3.4.4.5 ROC AUC Score

The ROC AUC score gives the probability that a randomly chosen positive instance (student is at risk) is higher than a randomly chosen negative instance (student is not at risk).

3.5 On the Credibility of the Study

The quality of research can be evaluated using two criteria, namely validity and reliability.

3.5.1 Validity

Validity checks if a measurement instrument or process measures what it is supposed to measure (Andrade, 2018; Leedy & Ormrod, 2019).

There are two types of validity namely internal and external. Internal validity refers to the research study and if it was a true reflection of what was studied (Punch, 1998). External validity refers to whether the findings of the study are able to be generalized to other contexts (Andrade, 2018; Punch, 1998).

Internal validity of this study will be achieved through the use of the ML pipeline, and the results that were obtained, by using four different metrics (accuracy, balanced accuracy, F1 score and ROC AUC score) to cross reference the results.

External validity (or generalizability) will be achieved through the use of unseen data for predictions during the ML process. If similar data is collected from a different source, the prediction scores are expected to be very similar to the current predictions.

3.5.2 Reliability

Reliability refers to the consistency of results produced by the measurement instrument (Leedy & Ormrod, 2019). Reliability in essence implies that if the same measurement instrument is used under the same conditions, but at different times, to what extent will it give the same results?

Since twenty-five MLAs will be used with forty ML models, reliability will be achieved by comparing the results of the models over 40 experiments, and observing the closeness of the prediction results.

3.6 Summary

This chapter covered the research paradigm for this study, which is post-positivism, the research strategy, which is quantitative, and the research method, which is the experimental method. The proposed experimental set up was also explained in detail, including the evaluation metrics that will be used to evaluate the prediction accuracy of the different models. The credibility of the study and how it will be evaluated was also discussed. The next chapter will provide a detailed discussion of the experiments conducted for this study using the methodology discussed in this chapter.

Chapter 4: Discussion on Conducting the Experiment

4.1 Introduction

This chapter presents the results of the experiments that used ML techniques to identify SAR. Some conclusions based on the findings will also be discussed. The first part discusses the dataset used, and the rationale for its use. The second part discusses the data preparation process, which is a necessary step before the data can be used for ML. This latter part delves into the details of data cleaning, feature selection and engineering, as well as feature scaling and selection. Twenty-five MLAs were applied to the processed data in order to identify SAR. This is an iterative process, whereby the models are evaluated and tuned in order to obtain satisfactory results. Different classification metrics were used in order to evaluate the performance of the MLAs to identify SAR. The coding for the experiments was done using Python V3.8.3 and Jupyter Notebook V6.0.3. In addition, several open-source Python libraries were used from the data cleaning, to the ML implementation and evaluation process. The code was run on an i7-8650U CPU with 8GB of RAM.

4.2 Data Wrangling and Processing

The quality and format of the raw student data is important to enable the ML models to make accurate predictions. The raw data requires pre-processing so it can be transformed into a form that can be used for ML. In real-life situations, the data can be quite messy and may thus require a lot of pre-processing before it can be used in ML. In a survey for Forbes of 80 data scientists about their jobs, Press (2016) reported on the following: Data scientists spent 60% of their time on the cleaning and organizing of the data. A further 19% of their time was spent on collecting data sets. According to that study, the data scientists spent close to 80% of their time on collecting, cleaning, and organizing the data. Data preparation is therefore one of the more challenging, but essential tasks in the ML *pipeline*. Figure 4.1 outlines the steps followed in this study to process the data for ML.

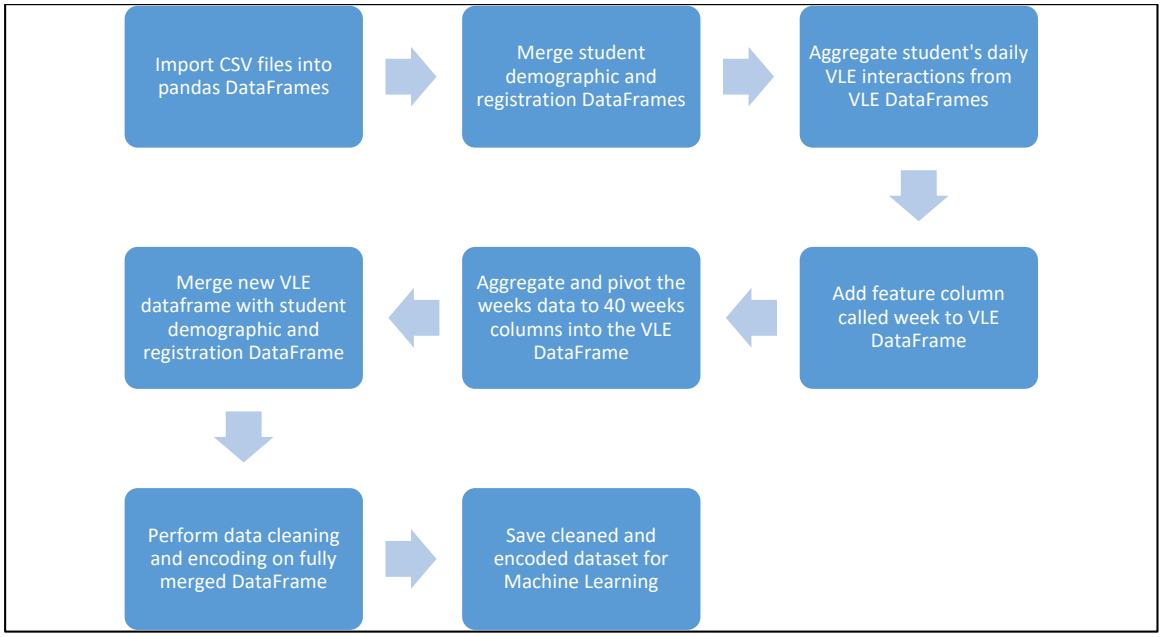

Figure 4.1: Data processing steps. Source: Researcher's own creation.

Data wrangling, which involves gathering, selecting, cleaning, and transforming data as part of the ML process, will be discussed in the next sections.

4.2.1 Data Acquisition and Description

The dataset for this study is made up of seven files in comma-separated values (CSV) format. Table 3.2 in chapter 3 provides a complete list of the attributes contained in each file. However, a summary describing each file is provided in Table 4.1 below.

Table 4.1: Summary of the different CSV files contained in the OULAD database.

CSV File	No. of Instances (Rows)	No. of Attributes (Columns)	Description
studentInfo	32 593	12	Student demographic information and final result for a course that was taken.
courses	22	3	The courses that were taken, when they started, as well as their duration in days.
studentRegistration	32 593	5	Details about the courses a student is registered for, as well as their registration date.
assessments	206	6	Assessments types and weights.
studentAssessment	173 912	5	Assessment scores from students.
studentVle	10 655 280	6	Every student interaction (called sum_click) with the VLE resources, including the number of times, as well as the relative date of the interaction.
vle	6364	6	The different types of VLE resources available.

Part of the pre-processing step involves importing the CSV files into two-dimensional data structures called DataFrames (DFs), containing rows and columns of data. This step is discussed in the next section.

4.2.1.1 Importing of CSV Files into DataFrames

Pandas is a Python data analysis package containing high-level data structures (such as DFs), that can be used for real-world data analysis of labelled data (Heydt, 2017; McKinney, 2015).Through the use of the *pandas* package, the CSV files mentioned in Table 4.1 were imported into DFs. The names of each resultant DataFrame (DF) are shown in Table 4.2.

Table 4.2: DataFrames created from the CSV files.

CSV File	DataFrame
studentInfo	dfstuI
Courses	dfcourse
studentRegistration	dfstuR
Assessments	dfassess
studentAssessment	dfstuA
studentVle	dfstuV
Vle	dfvle

4.2.1.2 DataFrames Visualization

Often, when working with multiple DFs, in order to analyse the data more effectively, it may help to visualize the data in the DFs. Visualizing the data may allow a researcher to gain greater insights into the data, in terms of its content and structure.

The following seven tables (Table 4.3 to Table 4.9) show a subset of each DF that was created from each of the seven CSV files.

Table 4.3: Subset of the dfstui DataFrame with students' demographic data and final result.

	code_module	code_presentation	id_student	gender	region	highest_education	imd_band	age_band	num_of_prev_attempts	studied_credits	disability	final_result
0	AAA	2013J	11391	M	East Anglian Region	HE Qualification	90-100%	55<=	0	240	N	Pass
1	AAA	2013J	28400	F	Scotland	HE Qualification	20-30%	35-55	0	60	N	Pass
2	AAA	2013J	30268	F	North Western Region	A Level or Equivalent	30-40%	35-55	0	60	Y	Withdrawn
3	AAA	2013J	31604	F	South East Region	A Level or Equivalent	50-60%	35-55	0	60	N	Pass
4	AAA	2013J	32885	F	West Midlands Region	Lower Than A Level	50-60%	0-35	0	60	N	Pass
...	...	...	...	...	...	...	...	...	...	...	...	...
32588	GGG	2014J	2640965	F	Wales	Lower Than A Level	10-20	0-35	0	30	N	Fail
32589	GGG	2014J	2645731	F	East Anglian Region	Lower Than A Level	40-50%	35-55	0	30	N	Distinction
32590	GGG	2014J	2648187	F	South Region	A Level or Equivalent	20-30%	0-35	0	30	Y	Pass
32591	GGG	2014J	2679821	F	South East Region	Lower Than A Level	90-100%	35-55	0	30	N	Withdrawn
32592	GGG	2014J	2684003	F	Yorkshire Region	HE Qualification	50-60%	35-55	0	30	N	Distinction

Table 4.4: Subset of dfcourse DataFrame with details about the courses taken.

	code_module	code_presentation	module_presentation_length
0	AAA	2013J	268
1	AAA	2014J	269
2	BBB	2013J	268
3	BBB	2014J	262
4	BBB	2013B	240
...	...	...	...
17	FFF	2013B	240
18	FFF	2014B	241
19	GGG	2013J	261
20	GGG	2014J	269
21	GGG	2014B	241

Table 4.5: Subset of dfstuR DataFrame with students' registration details.

	code_module	code_presentation	id_student	date_registration	date_unregistration
0	AAA	2013J	11391	-159	?
1	AAA	2013J	28400	-53	?
2	AAA	2013J	30268	-92	12
3	AAA	2013J	31604	-52	?
4	AAA	2013J	32885	-176	?
...	...	...	...	...	...
32588	GGG	2014J	2640965	-4	?
32589	GGG	2014J	2645731	-23	?
32590	GGG	2014J	2648187	-129	?
32591	GGG	2014J	2679821	-49	101
32592	GGG	2014J	2684003	-28	?

Table 4.6: Subset of dfassess DataFrame with the different types of assessments.

	code_module	code_presentation	id_assessment	assessment_type	date	weight
0	AAA	2013J	1752	TMA	19	10.0
1	AAA	2013J	1753	TMA	54	20.0
2	AAA	2013J	1754	TMA	117	20.0
3	AAA	2013J	1755	TMA	166	20.0
4	AAA	2013J	1756	TMA	215	30.0
...	...	...	...	...	...	...
201	GGG	2014J	37443	CMA	229	0.0
202	GGG	2014J	37435	TMA	61	0.0
203	GGG	2014J	37436	TMA	124	0.0
204	GGG	2014J	37437	TMA	173	0.0
205	GGG	2014J	37444	Exam	229	100.0

Table 4.7: Subset of dfstuA DataFrame with students' assessment scores.

	id_assessment	id_student	date_submitted	is_banked	score
0	1752	11391	18	0	78
1	1752	28400	22	0	70
2	1752	31604	17	0	72
3	1752	32885	26	0	69
4	1752	38053	19	0	79
...	...	...	...	...	...
173907	37443	527538	227	0	60
173908	37443	534672	229	0	100
173909	37443	546286	215	0	80
173910	37443	546724	230	0	100
173911	37443	558486	224	0	80

Table 4.8: Subset of dfstuV DataFrame with students' interactions with the VLE resources.

	code_module	code_presentation	id_student	id_site	date	sum_click
0	AAA	2013J	28400	546652	-10	4
1	AAA	2013J	28400	546652	-10	1
2	AAA	2013J	28400	546652	-10	1
3	AAA	2013J	28400	546614	-10	11
4	AAA	2013J	28400	546714	-10	1
...	...	...	...	...	...	...
10655275	GGG	2014J	675811	896943	269	3
10655276	GGG	2014J	675578	896943	269	1
10655277	GGG	2014J	654064	896943	269	3
10655278	GGG	2014J	654064	896939	269	1
10655279	GGG	2014J	654064	896939	269	1

Table 4.9: Subset of dfvle DataFrame with the different types of VLE resources available.

	id_site	code_module	code_presentation	activity_type	week_from	week_to
0	546943	AAA	2013J	resource	?	?
1	546712	AAA	2013J	oucontent	?	?
2	546998	AAA	2013J	resource	?	?
3	546888	AAA	2013J	url	?	?
4	547035	AAA	2013J	resource	?	?
...	...	...	...	...	...	...
6359	897063	GGG	2014J	resource	?	?
6360	897109	GGG	2014J	resource	?	?
6361	896965	GGG	2014J	oucontent	?	?
6362	897060	GGG	2014J	resource	?	?
6363	897100	GGG	2014J	resource	?	?

After inspecting the DFs shown, this researcher was able to gain a better understanding of how to perform the next part of the pre-processing step, which entailed the merging of the different DFs. This process is discussed in the next section.

4.2.2 DataFrames Merging

To perform data cleaning, the different DFs were merged into a single DF, through a series of steps. The process is outlined in the sections 4.2.2.1 to 4.2.2.3 that follow.

4.2.2.1 Merging Students' Demographic and Registration DataFrames

The dfstuI and dfstuR DFs both contain the same number of rows (32593). A comparison of the student_id column on both these DFs, through the use of the Python *assert* statement, showed an exact match of the rows. The dfstuI and dfstuR DFs were thus merged to create a new DF called dfstuDemo. The date_unregistration column was also dropped in the merging, since it was not required. The dfstuDemo DF, a subset of which is shown in Table 4.10, contains the students' demographics, as well as their registration details.

Table 4.10: The dfstuDemo DataFrame with students' demographic and registration details.

	code_module	code_presentation	id_student	gender	region	highest_education	imd_band	age_band	num_of_prev_attempts	studied_credits	disability	final_result	date_registration
0	AAA	2013J	11391	M	East Anglian Region	HE Qualification	90-100%	55<=	0	240	N	Pass	-159
1	AAA	2013J	28400	F	Scotland	HE Qualification	20-30%	35-55	0	60	N	Pass	-53
2	AAA	2013J	30268	F	North Western Region	A Level or Equivalent	30-40%	35-55	0	60	Y	Withdrawn	-92
3	AAA	2013J	31604	F	South East Region	A Level or Equivalent	50-60%	35-55	0	60	N	Pass	-52
4	AAA	2013J	32885	F	West Midlands Region	Lower Than A Level	50-60%	0-35	0	60	N	Pass	-176
...	...	...	...	...	...	...	...	...	...	...	...	...	...
32588	GGG	2014J	2640965	F	Wales	Lower Than A Level	10-20	0-35	0	30	N	Fail	-4
32589	GGG	2014J	2645731	F	East Anglian Region	Lower Than A Level	40-50%	35-55	0	30	N	Distinction	-23
32590	GGG	2014J	2648187	F	South Region	A Level or Equivalent	20-30%	0-35	0	30	Y	Pass	-129
32591	GGG	2014J	2679821	F	South East Region	Lower Than A Level	90-100%	35-55	0	30	N	Withdrawn	-49
32592	GGG	2014J	2684003	F	Yorkshire Region	HE Qualification	50-60%	35-55	0	30	N	Distinction	-28

The next step was to consider the dfstuV DF, a subset of which is shown in Table 4.8. This DF contains 10 655 280 rows of data. Based on the scope of this research, not all of the data in this the dfstuV DF was of interest in its current form. This researcher had to find some way of reducing the over 10 million rows of data to a more manageable number of rows, without simply deleting millions of rows of data. This was done through the use of data aggregation techniques. The next section discusses the data aggregation process as applied to the dfstuV DF.

4.2.2.2 Data Aggregation of Students' Virtual Learning Environment Data

Data aggregation involves deriving the correct level of detail in the data, so that the data can be used for ML. It involves combining a number of different data objects into a single object to create a new reduced dataset (Du, 2010). For this study, two levels of data aggregation were performed, namely daily then weekly.

- **Daily Data Aggregation of Students' Virtual Learning Environment Data**

Each time a student interacted with the different activities of the VLE, the details of that interaction were captured in an attribute called sum_click. These different types of activities were referenced from the activity_type attribute of the dfvle DF, as shown in Figure 4.5. On any given day, a student could have interacted with any number and type of VLE activities. Furthermore, the same student could have interacted with certain activities any number of times on any given day. Each row of data in the dfstuV DF reflected the total number of interactions with a specific activity type, by a specific student, in a specific course, on a specific day. In order to solve this problem, this researcher started by analysing the make-up of all the activity

types in the dfvle DF. Figure 4.2 shows a breakdown of the count of each type of activity in the dfvle DF.

resource	2660
subpage	1055
oucontent	996
url	886
forumng	194
quiz	127
page	102
oucollaborate	82
questionnaire	61
ouwiki	49
dataplus	28
externalquiz	26
homepage	22
glossary	21
ouelluminate	21
dualpane	20
repeatactivity	5
htmlactivity	4
sharedsubpage	3
folder	2

Figure 4.2: The different activities in the dfvle DataFrame.

There are twenty different activity types. Each of these activity types further contains a certain number of that specific activity type. This ranges from 2660 resource activities to 2 folders. A student on any given day, during any given time could have accessed any one of these different activity types. Accessing any of these activity types may not necessarily have any significant value on their own. This researcher considered that it may be more meaningful to aggregate a specific student's access to the different activities, using a time-based interval, such as days, as opposed to considering each activity accessed on their own. If we consider homepage access as an example, it is expected that in order for a student to access other VLE resources, the

student would have to access the homepage first. Thus, considering homepage access on its own may not necessarily provide any useful information. What may be more relevant is to get an overview of that student's access to the entire VLE within a given timeframe, such as a specific day. When a student interacts with a specific activity type, all interactions of that activity for that student doing a specific module, in a specific semester, on a specific day, are stored in the sum_click attribute for that particular student. This interaction is stored as one row of data, a subset of which is shown in Table 4.11, depicting the dfstuV DF. Instead of keeping each of these interactions for that particular student in separate rows, the sum_click values for that particular student were summed up. In effect, a particular student's entire interaction with the VLE for a specific course, in a specific semester, on a specific day, was summed up. The number of rows of data was thus reduced from 10 655 280 to 1 808 119. This resulted in a new DF called dfstuVSumD, as can be seen in Table 4.11, with a significantly fewer number of rows.

Table 4.11: The dfstuVSumD DataFrame with the aggregate sum_click per day.

	id_student	code_module	code_presentation	date	sum_click
0	6516	AAA	2014J	-23	28
1	6516	AAA	2014J	-22	82
2	6516	AAA	2014J	-20	41
3	6516	AAA	2014J	-17	7
4	6516	AAA	2014J	-12	2
...	...	...	...	...	...
1808114	2698588	BBB	2014J	240	3
1808115	2698588	BBB	2014J	244	3
1808116	2698588	BBB	2014J	248	5
1808117	2698588	BBB	2014J	250	2
1808118	2698588	BBB	2014J	258	1

The dfstuVSumD DF contains the aggregated sum_click values for a student who has interacted with the VLE on a specific day, as per the date attribute. It thus shows a student's overall interactions with the VLE per day.

Due to the extensive anonymization of the OULAD data, dates were converted to numbers representing the day number of the module, from the start of that module. Since students were allowed to access the VLE a few weeks before a module started, negative numbers in the date attribute represented the number of days before the module started. Figure 4.3 shows the start (-25) and end (269) dates in days, of the students' interactions with the VLE.

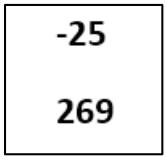

Figure 4.3: Start and end dates in days, of the VLE interactions.

The earliest interaction with the VLE was on day number -25. This means that a student first accessed a module on the VLE 25 days before the course started. Since the dfstuVSumD DF is ordered on the id_student attribute, and not on the date attribute, the first row in Figure 4.10 shows a date of -23 and not -25. Similarly, the final row shows a date of 258 and not 269.

- **Weekly Data Aggregation of Students' Virtual Learning Environment Data**

While the dfstuVSumD DF contained significantly fewer rows after aggregating the sum_click values per day, it was still a large enough number to impact on the computation time and memory requirements. Since this study was concerned with identifying SAR, this researcher needed to consider the factors that may help to do this. One of the factors was the students' interaction with the VLE. This researcher concluded that a student should not be considered to be classified as being at risk simply because that student may not have interacted with the VLE on certain days. There could have been any number of reasons for students to have not interacted with the VLE on any specific day. However, if a student had minimal or no interaction with the VLE for a longer time period, such as a week, it could provide more meaningful data to help determine whether that student is at risk or not. Thus, the next part of the discussion focuses on the process of aggregating the sum_click values from daily to weekly values. The other advantage of aggregating the sum_click values in this way was to help reduce the size of the DF, thus reducing the computation time and memory requirements.

In order to aggregate the values from daily to weekly values, a new attribute column called week was created in the dfstuVSumD DF. For each row of data in this DF, the value in the

date attribute column representing a day number, was converted to a value representing a week number, which was then stored in the corresponding week attribute column. Table 4.12 shows the result of this process.

Table 4.12: The dfstuVSumD DF showing the new week attribute column.

	id_student	code_module	code_presentation	date	sum_click	week
0	6516	AAA	2014J	-23	28	0
1	6516	AAA	2014J	-22	82	0
2	6516	AAA	2014J	-20	41	0
3	6516	AAA	2014J	-17	7	0
4	6516	AAA	2014J	-12	2	0
...	...	...	...	...	...	...
1808114	2698588	BBB	2014J	240	3	35
1808115	2698588	BBB	2014J	244	3	35
1808116	2698588	BBB	2014J	248	5	36
1808117	2698588	BBB	2014J	250	2	36
1808118	2698588	BBB	2014J	258	1	37

For the purpose of this study, all dates before a module started were aggregated and represented as week 0, which is the week just before the module started. The rest of the weeks was calculated according to the actual day values. Figure 4.4 shows the start (0) and end (39) weeks of the students' interaction with the VLE.

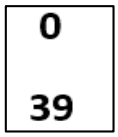

Figure 4.4: Start and end Weeks, of the VLE interactions.

The next step involved aggregating the sum_click values on a weekly basis, per student. The *pandas pivot_table()* function was used to rearrange the aggregated weekly sum_click values in each row, into weekly columns from 0 to 39, for each of the 40 weeks. Table 4.13 shows the resultant DF called dfstuVSumW, after applying the *pivot_table()* function on the dfstuVSumW DF.

Table 4.13: Multi-index dfstuVSumW DataFrame with aggregated sum_click values for 40 weeks.

		week	0	1	2	3	4	5	6	7	8	9	...	30	31	32	33	34	35	36	37	38	39
id_student	code_module	code_presentation																					
6516	AAA	2014J	256	229	42	79	193	69	34	10	93	57	...	90	55	54	119	79	67	0	0	0	2
8462	DDD	2013J	81	81	146	9	23	79	18	63	17	10	...	0	0	0	0	0	0	0	0	0	0
		2014J	0	0	10	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
11391	AAA	2013J	98	183	20	100	0	26	60	22	20	0	...	0	43	35	0	15	85	7	1	0	0
23629	BBB	2013B	14	23	5	9	2	0	8	0	23	35	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2698257	AAA	2013J	24	58	10	124	75	33	16	60	46	18	...	21	25	1	9	21	0	0	0	0	1
2698535	CCC	2014B	28	150	217	6	0	19	0	2	6	4	...	0	0	0	0	0	0	0	0	0	0
	EEE	2013J	22	17	2	205	69	485	284	409	234	35	...	2	7	0	27	70	0	0	0	0	0
2698577	BBB	2014J	0	0	0	49	23	96	150	12	1	12	...	1	3	0	0	19	0	0	0	0	0
2698588	BBB	2014J	0	9	72	18	0	7	2	7	18	0	...	0	25	17	13	147	12	7	1	0	0

The dfstuVSumW DF contains the weekly sum_click values per student, taking a specific module in a specific semester, over 40 weeks. These numbers represent, on a week-by-week basis, a student's interaction with the VLE for a specific module in a specific semester. All missing values were replaced with 0, since the student did not interact with the VLE for that week. The number of rows has been reduced from the original 10 655 280 rows in the studentVle DF, to 1 808 119 in the dfstuVSumD DF, and finally down to 29 228 rows in the dfstuVSumW DF. The dfstuVSumW DF is a *pandas* multi-index hierarchical DF, which needed to be flattened to a single index DF. This was done using the *pandas to_records()* function as shown in Table 4.14.

Table 4.14: Flattened dfstuVSumW DataFrame with aggregated sum_click values for 40 weeks.

	id_student	code_module	code_presentation	0	1	2	3	4	5	6	...	30	31	32	33	34	35	36	37	38	39
0	6516	AAA	2014J	256	229	42	79	193	69	34	...	90	55	54	119	79	67	0	0	0	2
1	8462	DDD	2013J	81	81	146	9	23	79	18	...	0	0	0	0	0	0	0	0	0	0
2	8462	DDD	2014J	0	0	10	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	11391	AAA	2013J	98	183	20	100	0	26	60	...	0	43	35	0	15	85	7	1	0	0
4	23629	BBB	2013B	14	23	5	9	2	0	8	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
29223	2698257	AAA	2013J	24	58	10	124	75	33	16	...	21	25	1	9	21	0	0	0	0	1
29224	2698535	CCC	2014B	28	150	217	6	0	19	0	...	0	0	0	0	0	0	0	0	0	0
29225	2698535	EEE	2013J	22	17	2	205	69	485	284	...	2	7	0	27	70	0	0	0	0	0
29226	2698577	BBB	2014J	0	0	0	49	23	96	150	...	1	3	0	0	19	0	0	0	0	0
29227	2698588	BBB	2014J	0	9	72	18	0	7	2	...	0	25	17	13	147	12	7	1	0	0

Since this study entailed predicting SAR on a weekly basis, the ML modelling process involved forty iterations. The dataset for any given week comprised students' demographic data, as well as all weekly VLE data from week 0, up to and including the week for which one wants to predict SAR.

4.2.2.3 Merging Demographic and Weekly Virtual Learning Environment DataFrames

Through a series of steps, this researcher created the dfstuDemo DF, which contained students' demographic and registration data. The dfstuVSumW DF, which contained the students' weekly interactions with the VLE, was also created. To obtain the final DF on which to apply the data cleaning process, these two DFs were merged. This resulted in the dfFinalCumU DF, as shown in Table 4.15.

Table 4.15: Subset of dfFinalCumU DataFrame with demographic and weekly VLE data.

	code_module	code_presentation	id_student	gender	region	highest_education	imd_band	...	33	34	35	36	37	38	39
0	AAA	2013J	11391	M	East Anglian Region	HE Qualification	90-100%	...	0	15	85	7	1	0	0
1	AAA	2013J	28400	F	Scotland	HE Qualification	20-30%	...	12	0	1	0	0	0	0
2	AAA	2013J	30268	F	North Western Region	A Level or Equivalent	30-40%	...	0	0	0	0	0	0	0
3	AAA	2013J	31604	F	South East Region	A Level or Equivalent	50-60%	...	85	5	101	17	1	2	0
4	AAA	2013J	32885	F	West Midlands Region	Lower Than A Level	50-60%	...	27	25	3	7	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
29223	GGG	2014J	2640965	F	Wales	Lower Than A Level	10-20	...	0	0	0	0	0	0	0
29224	GGG	2014J	2645731	F	East Anglian Region	Lower Than A Level	40-50%	...	2	1	2	1	0	0	0
29225	GGG	2014J	2648187	F	South Region	A Level or Equivalent	20-30%	...	4	0	0	0	0	0	0
29226	GGG	2014J	2679821	F	South East Region	Lower Than A Level	90-100%	...	0	0	0	0	0	0	0
29227	GGG	2014J	2684003	F	Yorkshire Region	HE Qualification	50-60%	...	0	0	0	0	0	0	0

The dfFinalCumU DF contained all the necessary columns of data that were required for the ML process. However, before the data could be used for ML, this researcher had to perform data cleaning and transformation on the data contained in the dfFinalCumU DF. This process is discussed in the next section

4.2.3 Data Cleaning, Transformation and Feature Engineering

The resultant DF dfFinalCumU contains the combined columns of data that will be used for the next major steps, which are data cleaning, transformation, and feature engineering. Data cleaning involves filtering out irrelevant data, such as missing records, duplicated records, and incorrectly formatted data. Inaccurate data that is input to the ML models can result in inaccurate predictions and decision making. As indicated in section 3.4 of chapter 3, identifying SAR using ML is a binary classification, supervised ML task. For this study, ML models were

trained on a set of cleaned student data. The trained models could then be used to make predictions in order to identify SAR.

Table 4.15 depicts the dfFinalCumU DF with different types of data in the various columns. In general, a dataset can contain missing values, incorrect types of values, and outliers. These will need to be fixed either by correcting the errors, or deleting the rows (or columns) containing the invalid data.

A dataset that no longer has these errors will still require further processing before it can be used for ML. To perform optimally, MLAs, in general, require a dataset to contain only numeric values.

A series of steps is needed to be followed to transform the dfFinalCumU DF, as displayed in Table 4.15, into one that would be suitable for ML. A summary of these steps is depicted in Figure 4.5 below.

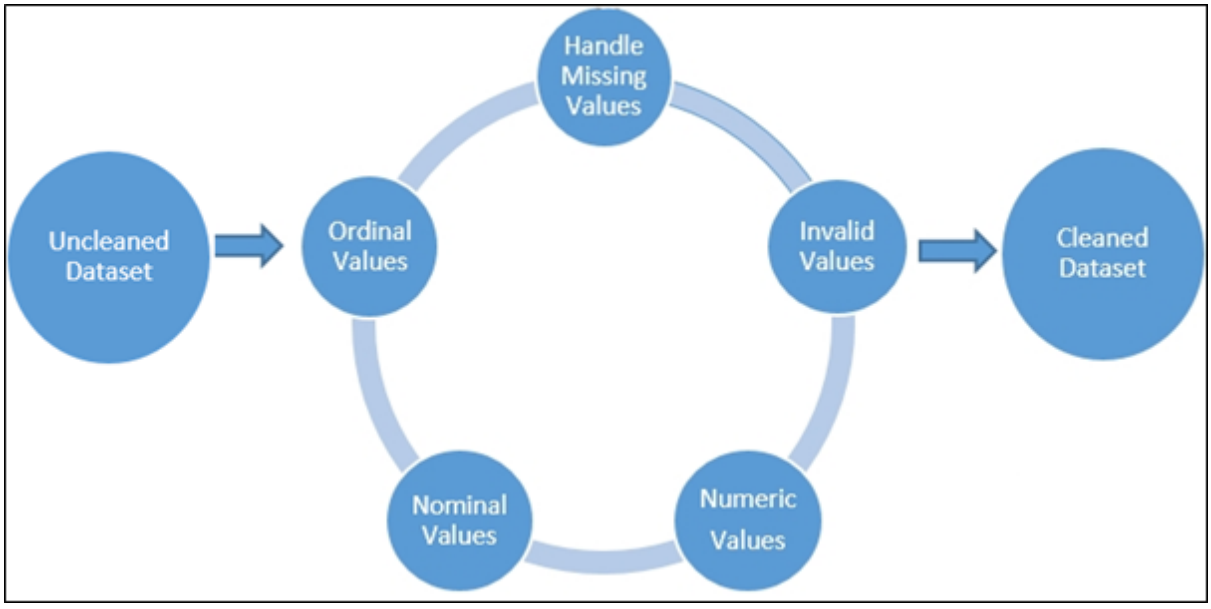

Figure 4.5: Summary of steps followed to transform the dataset for machine learning. Source: Researcher's own creation.

In order to perform the data cleaning and transformation, a Python function called *transform_data*() was written. The code listing of the function is shown in Figure 4.6. The steps followed in the function will be discussed in detail in the sections that follow.

```
def transform_data(df):
    nominalcols = ['code_module', 'code_presentation', 'region', 'disability', 'gender']
    numericcols = ['id_student','num_of_prev_attempts', 'studied_credits', 'date_registration']
    weekcols = list(map(str,[*range(0, 40, 1)]))
    ordinalcols = ['age_band','highest_education', 'imd_band']
    resultcol =  ['final_result']

    dfNominal = pd.get_dummies(df[nominalcols])

    df.drop(df[df.date_registration == '?'].index, inplace = True)

    dfNumeric = df[numericcols]

    df.loc[:,'imd_band'] = df['imd_band'].replace({'%': ''}, regex = True)

    df.replace({'?':np.nan},inplace = True)
    dfNulls = df.isnull().sum()

    for i in ordinalcols :
        df.loc[:,i] = df.loc[:,i].astype('category')
        df.loc[:,i] = df.loc[:,i].cat.codes
    dfOrdinal = df[ordinalcols]

    dfWeeks = df[weekcols]

    dfResult = df[resultcol].replace({'Fail':0,'Withdrawn':0,'Pass':1,'Distinction':1},inplace= False)
    dfResult.rename(columns={'final_result': 'risk_status'}, inplace=True)

    dfTransformed = dfNumeric.join(dfNominal).join(dfOrdinal).join(dfWeeks).join(dfResult)
    dfTransformed.dropna(how='any',inplace = True )
    dfTransformed.drop_duplicates(subset='id_student',keep = 'first', inplace = True)
    dfTransformed.reset_index(drop=True, inplace = True)

    return dfTransformed, dfNulls
```

Figure 4.6: Code listing of the transform_data() function.

4.2.3.1 Missing Data

Datasets can often have missing values for various reasons, such as through errors from the original input or data collection. Joining two datasets together can also result in unmatched values. These missing values can either be replaced with other values, such as the mean value, using techniques such as data imputation, or even deleted from the dataset entirely. Missing values are handled differently, depending on the individual dataset. Figure 4.7 shows the number of missing values in each column of the dfFinalCumU DF.

```
imd_band              1054
date_registration        7
```

Figure 4.7: Missing values in the dfFinalCumU DataFrame.

Based on an analysis of the dataset and its large size, this researcher chose to delete the rows containing the missing values.

Missing values in the original CSV files were displayed with a question mark, which got transferred to the DFs. The code in lines 10, 16 to 17, and 30 of the code listing depicted in Figure 4.6, handles the missing values. Question marks were replaced with null values. Rows containing null values were then dropped from the DF.

4.2.3.2 Invalid or Inconsistent Data

There will often be different types of data in a dataset. However, there should be consistency in the representation of data within the columns. There may be characters in columns where there should only have numeric values, for instance. Some data in a column may not follow an expected pattern, based on the rest of the data in that column. These types of anomalies need to be fixed, either through deletion or correction, depending on the type of anomaly. One of the anomalies identified in the dataset was in the imd_band column. The column has percentage intervals ranging from 0-10% all the way up to 90-100%. However, the second interval from the original data entry was stored as 10-20, instead of 10-20% as expected. This anomaly was fixed by removing the % character from all the values in imd_band column, through the use of a matching regular expression pattern, as shown in line 14 of the code listing in Figure 4.6. After this fix, the values in this column followed a consistent pattern.

4.2.3.3 Numerical Data

ML models require data to be in numerical format to perform optimally. Thus, unless the numerical values require further processing, such as scaling or normalizing, variables that are in numerical format already, may not require any further changes. Lines 3-4 of the code listing in Figure 4.6 show the numerical columns, while lines 10 and 24 show these columns being stored in the DFs, without any changes. Variables that are not in numerical format will be further processed to convert them to numerical format. This process will be discussed in the following sections.

4.2.3.4 Categorical Data

Categorical data is non-numeric data representing the characteristic of an object, such as the gender of a person. There are no numeric values attached to the categories, and as such, mathematical calculations cannot be applied to categorical data. Categorical data falls into two sub-categories, namely nominal and ordinal.

- **Nominal Data**

Data in the nominal category defines the category of data, but there is no natural ordering of the data. The gender of a person could be male or female, for instance. However, there is no

order that can be defined on values in this category. The dataset contains a number of nominal variables, as depicted in line 2 of the code listing shown in Figure 4.30. Since the data needed to be in numerical format, a technique called *one hot encoding* was applied to the nominal variables. This was done through the use of the *pandas get_dummies()* function, which converted the nominal variables to dummy variables. Line 8 of the code listing displayed in Figure 4.6, shows the implementation of the 'one hot encoding' technique on all of the nominal variables. Thus, a nominal variable such as gender, for instance, with two possible values F or M, was converted into two separate variables called gender_F and gender_M. If the gender of the first student was M (for male), then after the *get_dummies()* function was applied to the gender variable, there will be a 0 in the gender_F column and a 1 in the gender_M column in the first row. Figure 4.8 depicts this process for the first five students in the dataset.

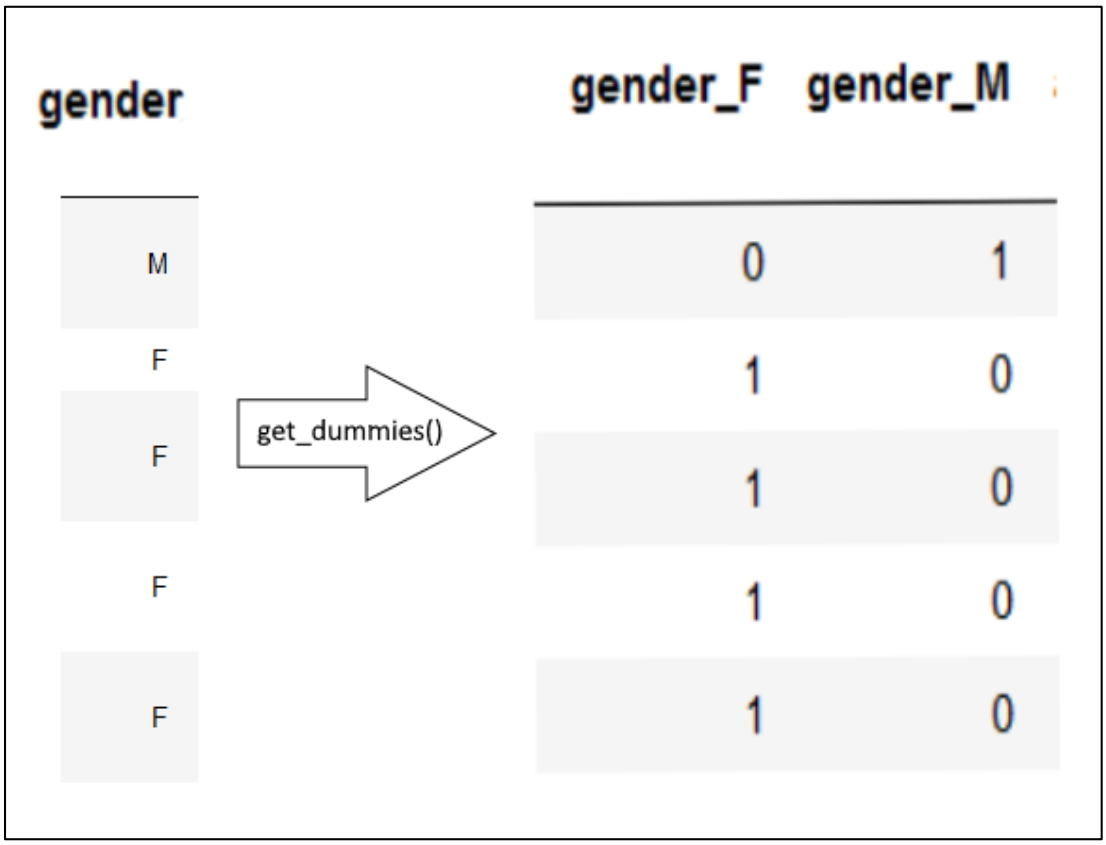

Figure 4.8: 'One hot encoding' of gender variable.

The same process was used to convert all the categorical variables to *one hot encoded* variables.

- **Ordinal Data**

Data in the ordinal category is also considered categorical data. However, ordinal data can be ordered into numeric categories, starting from 0 and increasing by 1 for each new sub-category

of value that a specific variable may contain. Our dataset contains three ordinal variables as shown in line 5 of the code listing displayed in Figure 4.6. The data in each of these categories of variables is converted to numeric values. An example of this conversion for the first five rows of the imd_band variable is depicted in Figure 4.9. Lines 19 to 22 of the code listing displayed in Figure 4.6 show the implementation of encoding the ordinal variables into numeric categories.

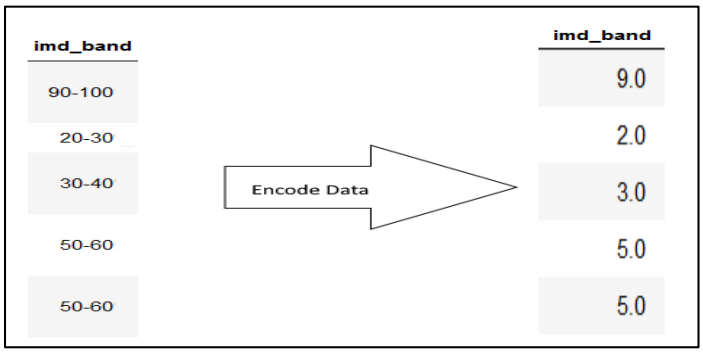

Figure 4.9: Numeric category encoded imd_band.

4.2.3.5 Target Variable Re-encoding

Students' final results were based on two tests and a final examination. The final_result variable contains four types of results: pass, distinction, withdrawn, and fail. Since the objective is to be able to identify SAR using binary classification ML techniques, the final_result values were re-encoded from the four string values to binary values, with a value of 0 identifying a student at risk, and a value of 1 identifying a student who is not at risk. The final_result variable was also renamed to risk_status. The re-encoded values were stored in the risk_status variable. A student who failed or withdrew from their course, was encoded with a risk_status of 0, and a student who passed, or passed with a distinction, was encoded with a risk_status of 1. Since the target variable was originally based on the students' test and examination marks, it meant that there would be a high correlation between the test and exam variables, with that of the final_result variable. As such, these variables were not used to identify SAR. While this may seem counter-intuitive, in order for the MLAs to perform optimally and to avoid overfitting, these variables were not considered. Lines 26 to 27 of the code listing displayed in Figure 4.6, shows the implementation of re-encoding the target variable.

4.2.3.6 Duplicate Data

Duplicate data can lead to overfitting during the training of the ML models. After the necessary transformation of the data has taken place, all duplicate data was deleted from the dataset. Lines

31 to 32 of the code listing depicted in Figure 4.6, show the implementation of de-duplicating the data.

4.3 Exploratory Data Analysis

The exploratory data analysis (EDA) step helps one to understand and gain insights into the dataset through intuitive techniques, such as visualization, as well as rigorous techniques, such as statistics. This allows one to discover any patterns or even anomalies in the data, before any ML techniques are applied to the data. In this section we explore some of the features of the dataset and make some inferences about the dataset. Categorical features will be broken down according to the number of students who passed, and the number who failed, in the different categories that are being explored.

4.3.1 Descriptive (Univariate) Analysis

Univariate analysis of the data allows one to gain a better understanding of the characteristics of the individual attributes or features in the dataset. The next section discusses the univariate analysis process. This process can assist with selecting the appropriate features for ML.

4.3.1.1 Outlier Detection Using Boxplots

Outliers are observations in the dataset that are much further way, or an abnormal distance, from other data points in a random sample of values. Part of the EDA process is to determine whether outliers are anomalies that need to be removed, or actual valid values. Some MLAs perform better with outliers removed. A boxplot depicts a distribution of data at different quartiles. This can help to identify potential outliers in the dataset. Figure 4.10 depicts boxplots of categorical features of the student dataset. Figure 4.11 depicts boxplots of forty weeks of students' interactions with the VLE.

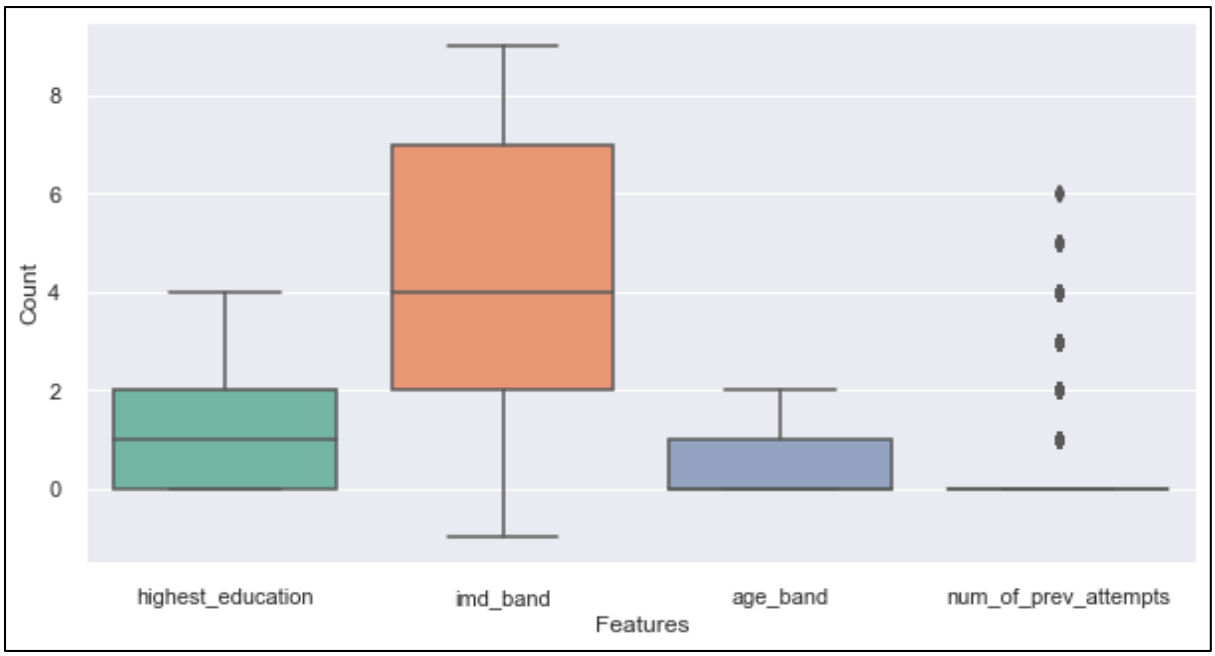

Figure 4.10: Boxplots of categorical features.

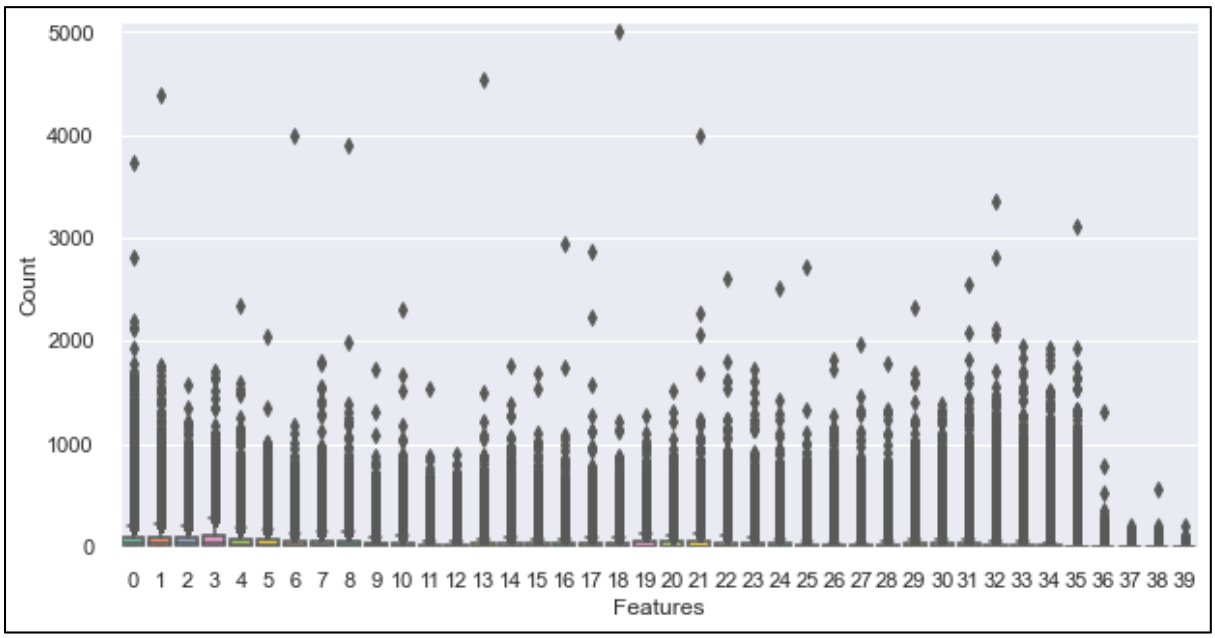

Figure 4.11: Boxplots of weekly VLE interactions.

From the two boxplots, num_of_prev_attempts has potential outliers, since it is unusual for students to attempt a course as many as six times. From Figure 4.11, it can be observed that the values are within an acceptable range and will not be regarded as outliers.

4.3.1.2 Countplots

Countplots show a breakdown of categorical variables. A count of observations in each category is shown using bar graphs.

- **Gender and Final Result**

Figure 4.12 shows a breakdown of students' results according to gender. There is a fairly even split in the number of male students who passed, and those who failed. In terms of female students, there was a higher number who passed, than those who failed. Overall, there were more male students than female ones.

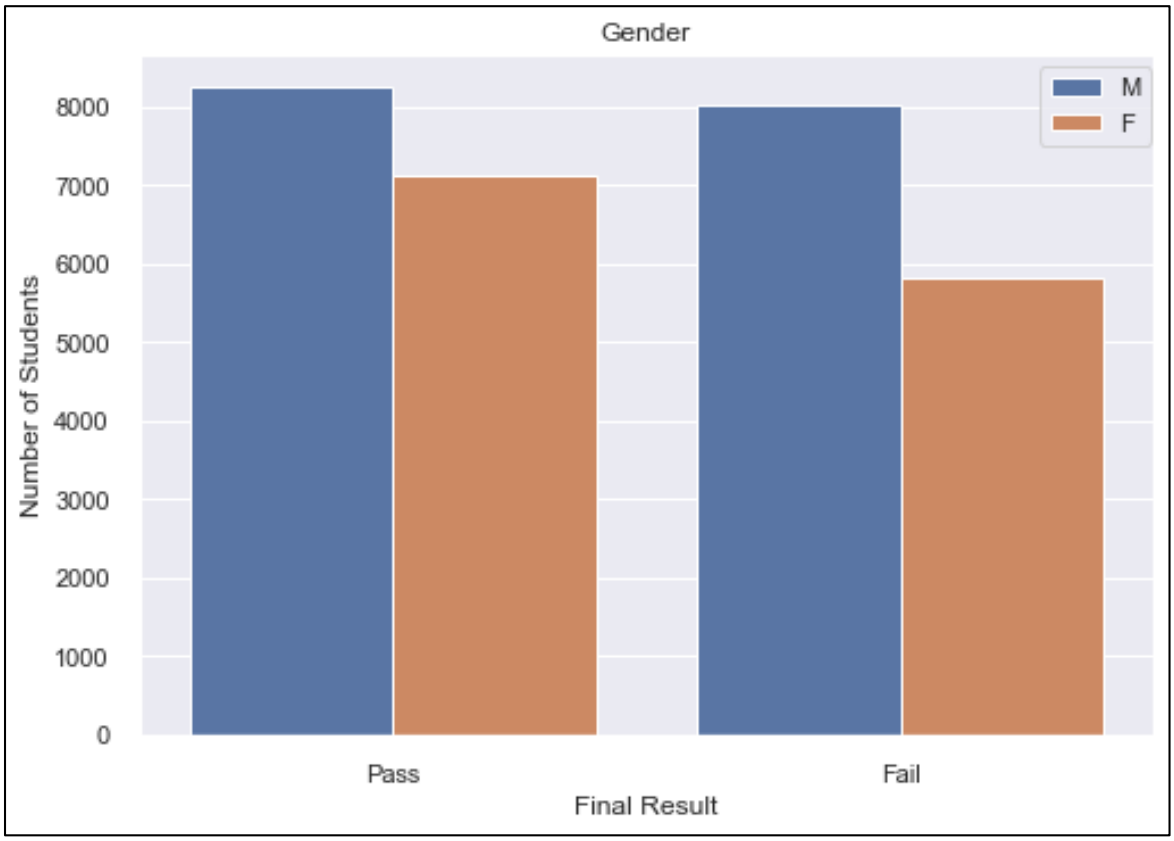

Figure 4.12: Students' results according to gender.

- **Age Band and Final Result**

Figure 4.13 shows a breakdown of students' results according to gender. Most of the students were under 35 years of age. This is to be expected for students attending university. There was also a fairly even split in the pass and failure rate among students in this age group. There was a very small number of students that was 55 years and over.

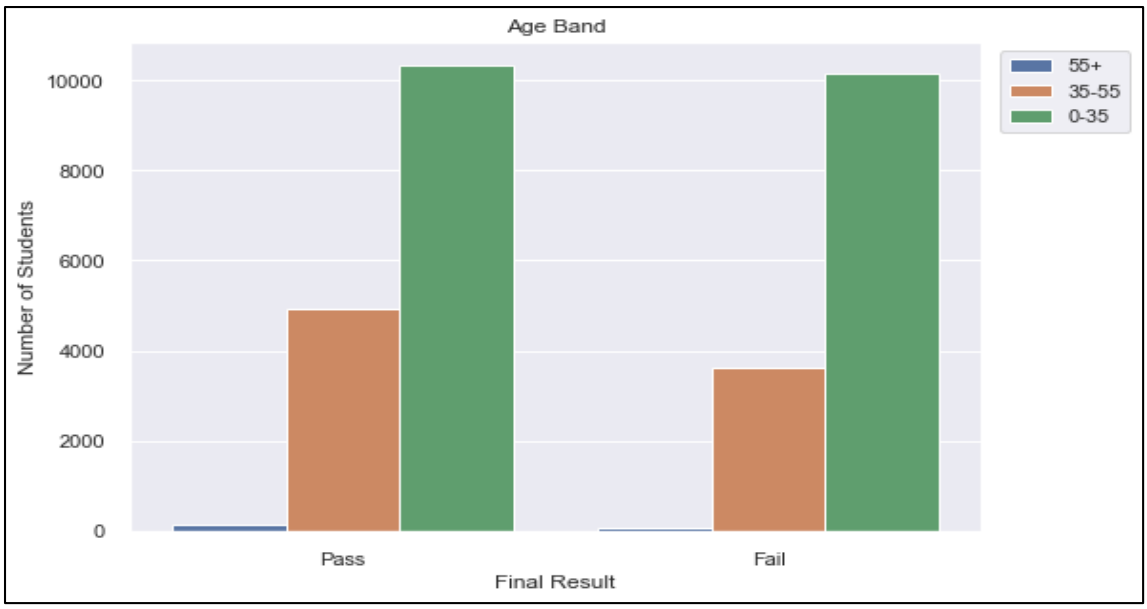

Figure 4.13: Students' results according to age band.

- **Highest Education and Final Result**

Figure 4.14 shows a breakdown of students' results according to highest education already obtained. There was a higher pass rate among students who obtained previous certificated qualifications, or those who took courses in specific subject areas prior to entering the university. There was a very small number of students with postgraduate qualifications and those who entered without prior formal qualifications. Those students who entered the university from school had a much higher failure rate. This observation may be useful towards identifying SAR.

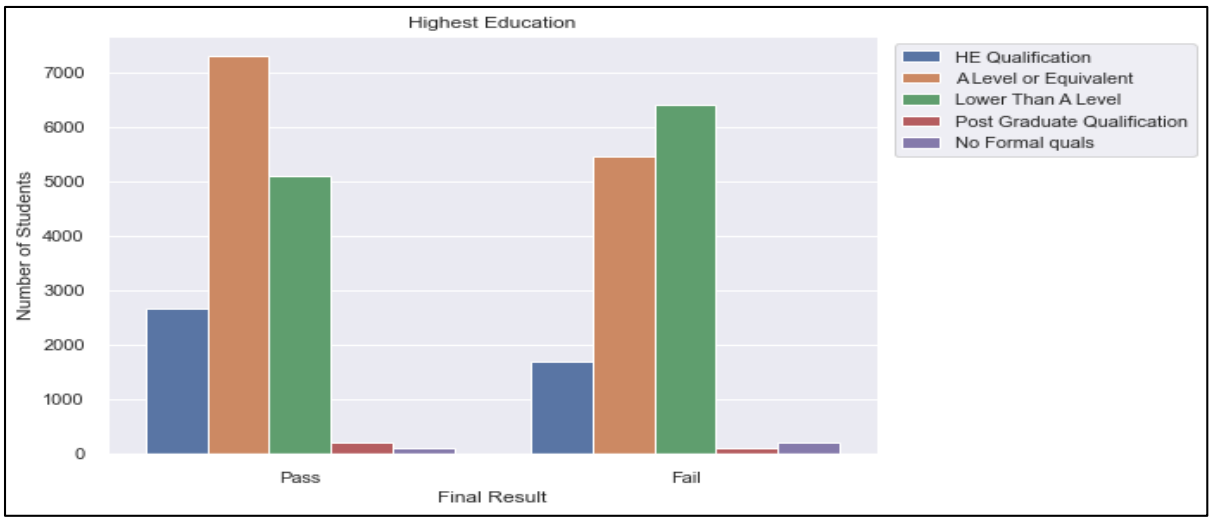

Figure 4.14: Students' results according to highest education.

- **IMD Band and Final Result**

Figure 4.15 shows a breakdown of students' results according to the IMD band. The IMD band, or Index of Multiple Deprivation Band, measures how deprived an area is (Bowie, 2019). This ranges in groups of 10%. The most deprived people would thus be in the range of 0%-10% and the least deprived would be those people in the range of 90%-100%. The graph shows a high failure rate among those students living in the most deprived areas. It can also be observed from the graph that there are some students who are not categorized in any of the ten IMD bands (as depicted by a question mark in the legend of Figure 4.15). These need to be investigated as they are most likely invalid data. There is also an inconsistent representation of the data in the 10-20 range. The missing percentage sign at the end, in this IMD band range, will result in an incorrect categorization of that range, if it is not fixed.

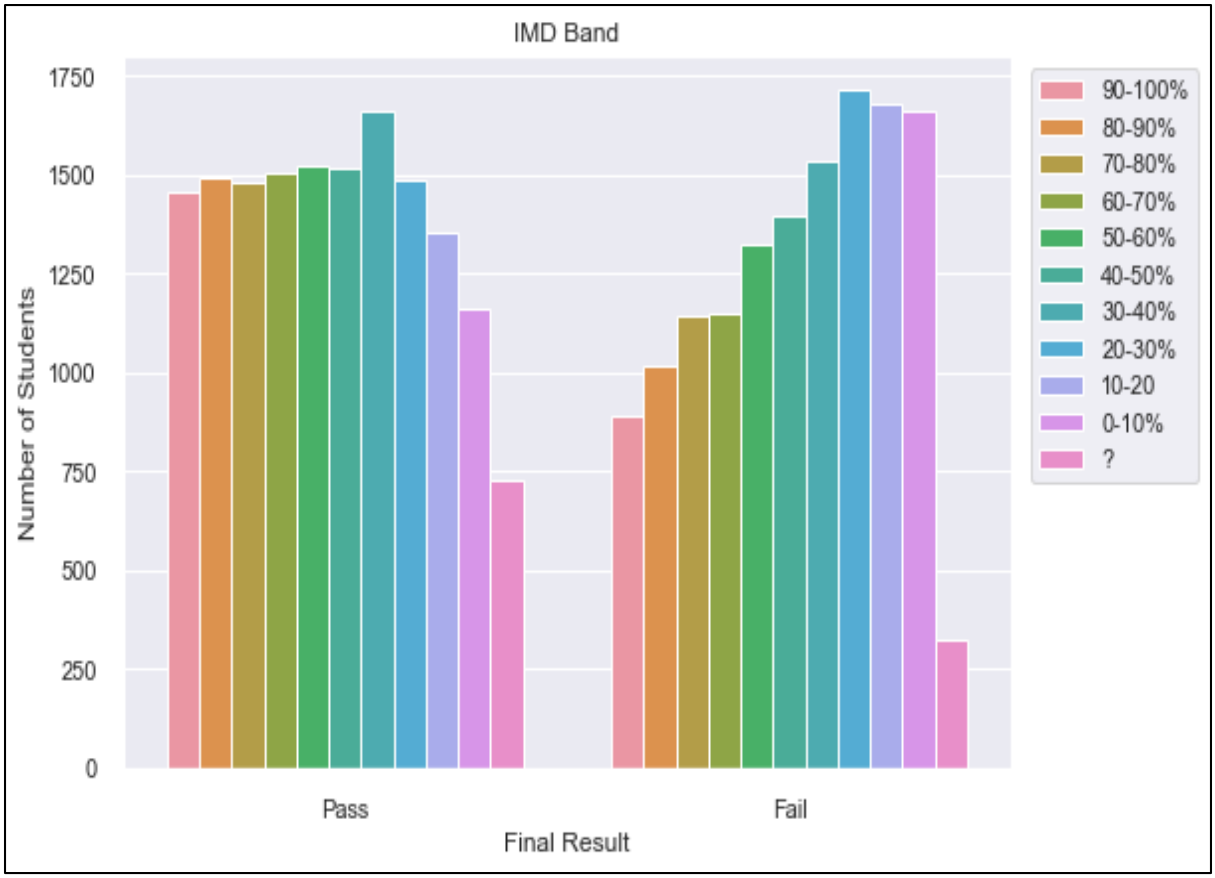

Figure 4.15: Students' results according to IMD band.

- **Code Presentation and Final Result**

Figure 4.16 shows a breakdown of students' results according to the code presentation, which is an identification code made up of the year, and a letter representing the semester, in which the courses were taken. Courses taken in semester one will contain the letter 'B' while courses taken in semester two will contain the letter 'J' in the code presentation. There was a fairly even split in the pass and failure rate among students who took their courses in the first semesters. There was a higher pass rate amongst those students who took their courses in the second semesters.

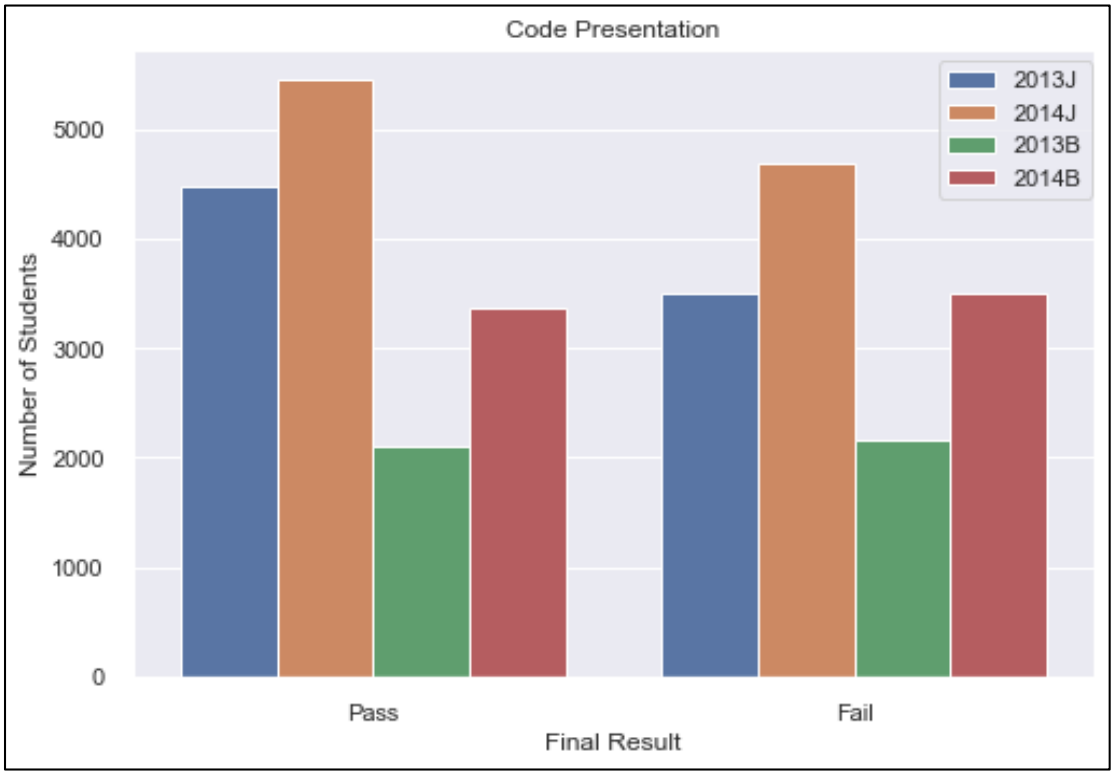

Figure 4.16: Students' results according to code presentation.

4.3.1.3 Pairplots

Pairplots show the distribution of the numerical variables. Pairwise plots between two variables allow for one to determine if there may be some kind of relationship between the pairs of data. Figure 4.17 shows the pairplots for the students' demographic data.

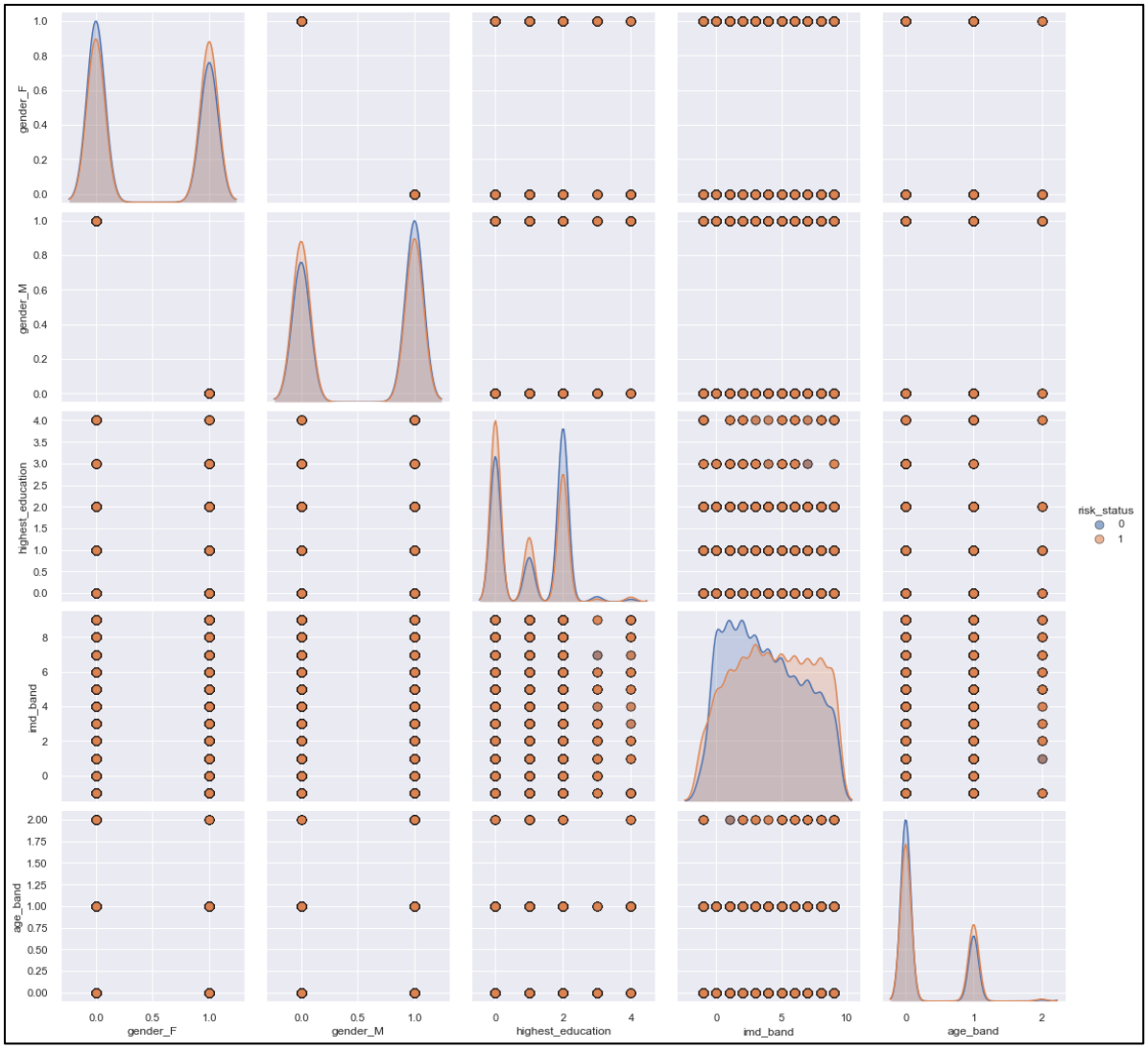

Figure 4.17: Pairplots for students' demographic data.

4.3.1.4 Correlation Analysis

A correlation between two variables concerns the strength of the relationship between their values (Bryman, 2012; Rowntree, 2004). When the value of one variable changes in a certain direction, so will the value of the other variable. Two highly correlated variables can add noise into the ML modelling process. The ML algorithm will learn from one variable and then may overfit, since the correlated variable will not add any new generalization capabilities (Darst, Malecki, & Engelman, 2018). A correlation matrix, or heat map, can be used to help determine if two or more variables are highly correlated. Figure 4.18 shows the correlation matrix using the Pearson Correlation Coefficient to determine if two or more variables are highly correlated. A highly positive correlation between two variables will be a number close to 1, while a highly

negative correlation will be a number close to -1 (Levine & Stephan, 2010). From the correlation matrix, it can be seen that there is not a high correlation between the variables.

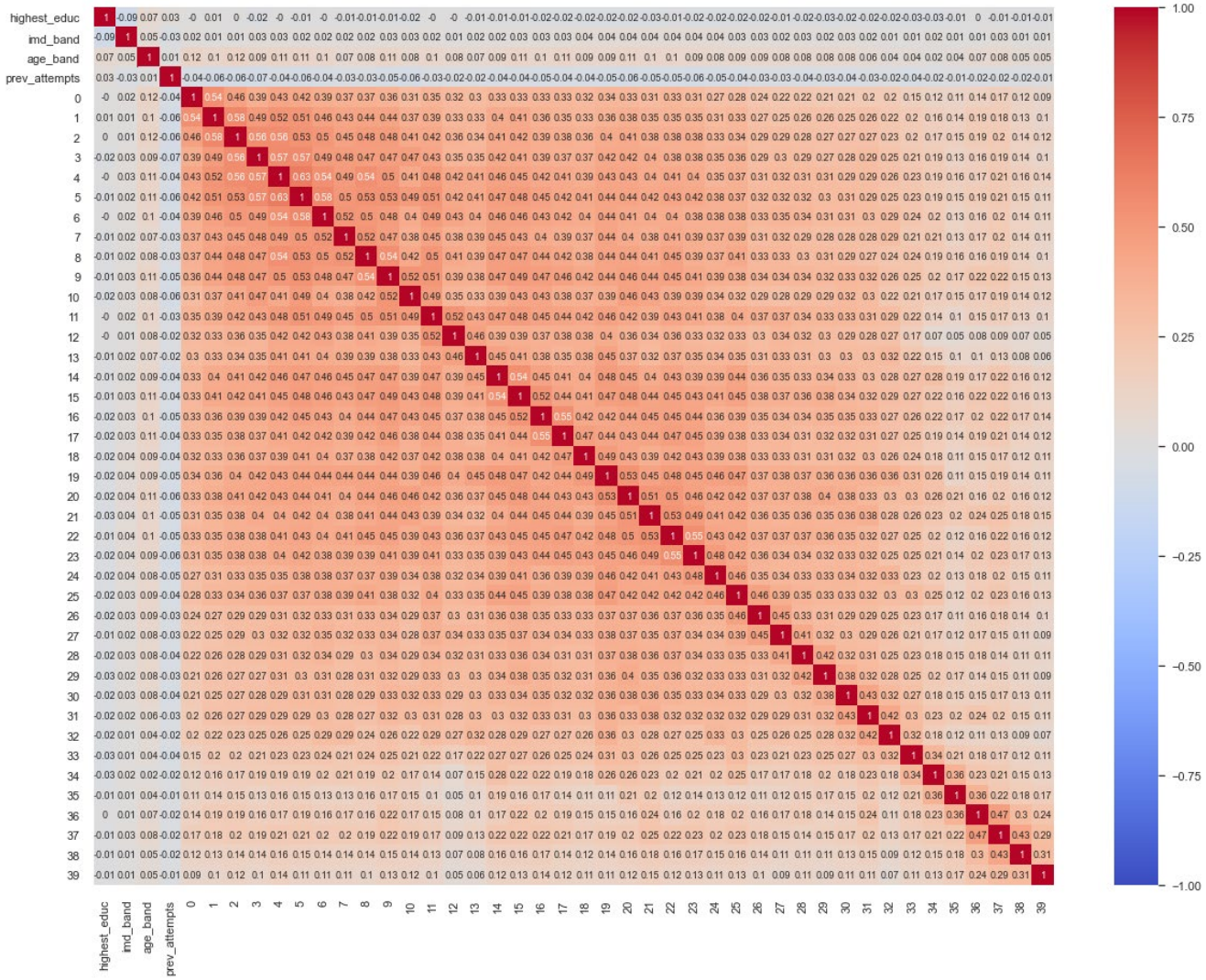

Figure 4.18: Correlation matrix for student's demographic and weekly VLE Data.

The resultant dataset from the data preparation and cleaning process discussed in the previous sections, is in a format that can be used for ML to be able to identify SAR. The following variables or features comprising demographic and weekly VLE data will be used as input to the ML models:

code_module_AAA, code_module_BBB, code_module_CCC, code_module_DDD, code_module_EEE, code_module_FFF, code_module_GGG, code_presentation_2013B, code_presentation_2013J, code_presentation_2014B, code_presentation_2014J, region_East Anglian Region, region_East Midlands Region, region_Ireland, region_London Region, region_North Region, region_North Western Region, region_Scotland, region_South East Region, region_South Region, region_South West Region, region_Wales, region_West

Midlands Region, region_Yorkshire Region, disability_N, disability_Y, gender_F, gender_M, age_band, highest_education, imd_band, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 ,19 ,20, 21, 22, 23, 24, 25 ,26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,37, 38, 39.

The values 0 to 39 represent the weekly column attributes of the VLE data. Thus, in order to train the ML model for the final week, for example, the inputs would comprise all of the demographic attribute data as shown, as well as the data in attribute 39.

The target class or output variable to be predicted by the model is:
risk_status.

4.4 Target Class and the Effect of Imbalanced Datasets

MLAs are designed to minimize error and maximize accuracy. If a dataset contains highly imbalanced target classes, the ML model can result in overfitting, whereby it will have a very high degree of accuracy in training, but predicts poorly on unseen data (Pes, 2020; Varmedja, Karanovic, Sladojevic, Arsenovic, & Anderla, 2019). In terms of binary classification, an imbalanced dataset will have a very high proportion of values from one class (the majority class), over the other class (Ali et al., 2019; Li et al., 2019; Priya & Uthra, 2021). The example that is often used in the literature is that of fraud detection in credit card transactions (Maniraj, Saini, Ahmed, & Sarkar, 2019; Varmedja et al., 2019; Yee, Sagadevan, & Malim, 2018). In general, with thousands of transactions, the percentage of fraud is very low. ML models will thus learn to predict, based on the dominant class of data, which says there is no fraud. This results in a high degree of accuracy during training. However, when given a fraudulent transaction to predict, the model tends to perform poorly, as it was unable to generalize. A balanced dataset is thus important for the ML models (Varmedja et al., 2019). A balanced dataset for a binary classification problem is one in which there is a fairly even split between the classes of the target variable. Figure 4.19 shows a fairly balanced dataset between the two classes, with 0 representing students who are at risk, and 1 representing students who are not at risk. Balancing of the classes was thus not required.

Figure 4.19: Showing a nearly balanced class distribution.

4.5 Feature Importance

Feature importance techniques help to understand how certain features contribute to predict the target variable. Figure 4.20 shows how each of the demographic features may influence the ML predictions.

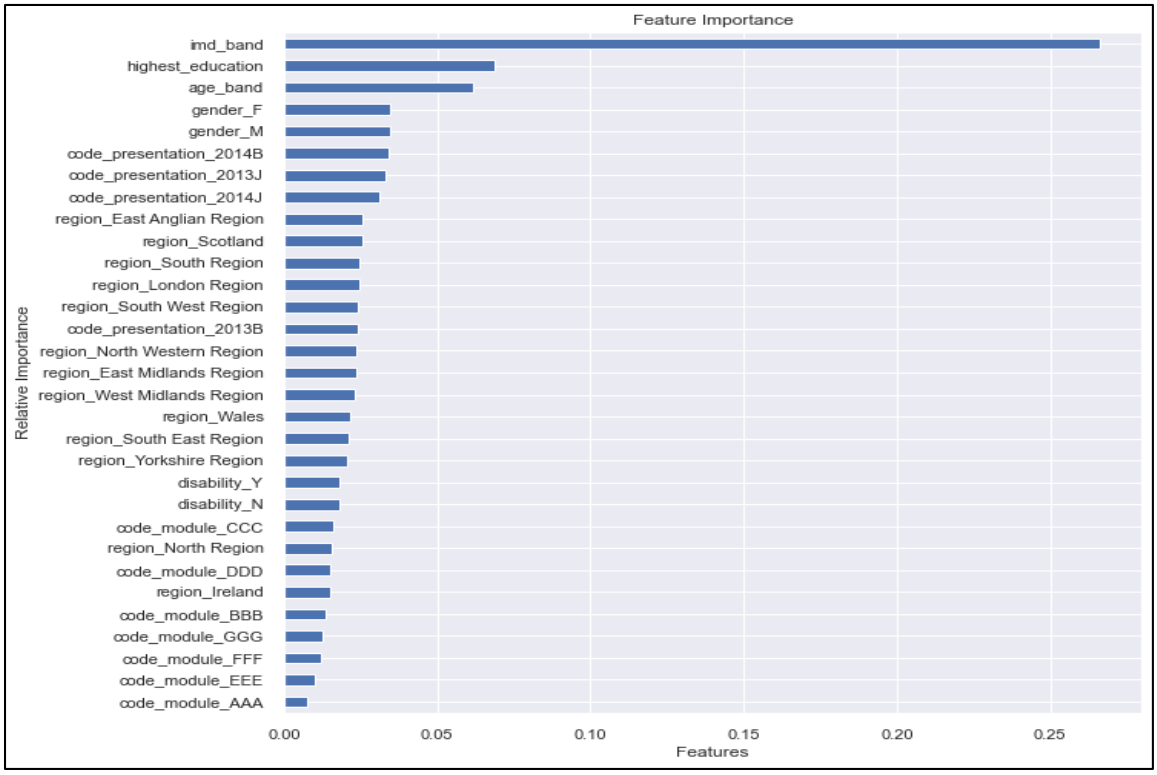

Figure 4.20: Showing demographic feature importance.

4.6 Description of the Machine Learning Model Performance (Building Predictive Models)

Evaluating the performance of a ML model is done using classification metrics. Some of the more common metrics such as accuracy, balanced accuracy, precision, recall, specificity, F1 score, and the ROC curve, explained previously, are revisited. The confusion matrix, from which several metrics are derived, is also explained for the SAR dataset.

4.6.1 Data Transforms

The dataset has different input variables, or feature columns, with varying ranges. If the input variables in one column, have much bigger values than those in other columns, it can affect the performance of MLAs. The algorithms may be biased towards the columns with the larger values resulting in poor ML performance. Algorithms such as logistic regression and artificial neural networks, that use a weighted sum of inputs, may be affected by this issue; so can algorithms, such as support vector machine and k-nearest neighbours, that use distance measures. It is thus preferable to scale or normalize the input data to a common range. The scaled data is then used for ML. For this study, the *MinMaxScaler()* function, from sklearn's pre-processing package, was used to scale the input data to a range between 0 and 1. The scaled dataset was used as input for the ML modelling process discussed in the next section.

4.6.2 Machine Learning Modelling Process and Evaluation Results

In order to evaluate the predictive capabilities of the MLAs, twenty-five different MLAs were used through a Python package called lazypredict (Pandala, 2020), which fits and evaluates all ML models from Python's scikit-learn package. Figure 4.21 shows the code listing for the ML modelling process that was used to generate performance metrics.

```
from lazypredict.Supervised import LazyClassifier
dfStuds = dfScaled
Y = dfScaled['risk_status']
dfModels = pd.DataFrame()
for i in range(1,40):
    dfStuds = dfStuds.iloc[:, :-1]
    X = dfStuds
    X_train, X_test, Y_train, Y_test = train_test_split (X, Y, test_size = 0.2, random_state = 0,stratify=Y)
    clf = LazyClassifier(verbose=0,ignore_warnings=True,custom_metric=confusion_matrix,classifiers='all')
    models, predictions = clf.fit(X_train, X_test, Y_train, Y_test)
    dfModels = dfModels.append(models)
```

Figure 4.21: Code listing for the machine learning modelling process.

Since there were 40 different sets of data, for each of the 40 weeks of the course, there were 40 iterations of the modelling process, as shown in Figure 4.21. As a consequence, there were 40 different sets of results output, with each set of results representing the modelling process for a particular week. Table 4.16 shows the results for a single week.

Table 4.16: Shows the results for a single week.

	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	confusion_matrix	Time Taken
Model						
AdaBoostClassifier	0.90	0.90	0.90	0.90	[[2252 242] [277 2444]]	2.13
BaggingClassifier	0.91	0.91	0.91	0.91	[[2185 309] [175 2546]]	3.17
BernoulliNB	0.80	0.81	0.81	0.80	[[2222 272] [762 1959]]	0.14
CalibratedClassifierCV	0.86	0.86	0.86	0.86	[[2271 223] [526 2195]]	16.98
DecisionTreeClassifier	0.87	0.87	0.87	0.87	[[2177 317] [386 2335]]	0.58
DummyClassifier	0.51	0.51	0.51	0.51	[[1202 1292] [1280 1441]]	0.10
ExtraTreeClassifier	0.84	0.84	0.84	0.84	[[2120 374] [486 2235]]	0.15
ExtraTreesClassifier	0.91	0.91	0.91	0.91	[[2108 386] [60 2661]]	2.96
GaussianNB	0.69	0.70	0.70	0.68	[[2311 183] [1413 1308]]	0.14
KNeighborsClassifier	0.77	0.77	0.77	0.76	[[2152 342] [883 1838]]	15.32
LGBMClassifier	0.93	0.92	0.92	0.93	[[2204 290] [97 2624]]	1.00
LinearDiscriminantAnalysis	0.81	0.81	0.81	0.81	[[2090 404] [599 2122]]	0.44
LinearSVC	0.86	0.86	0.86	0.86	[[2238 256] [463 2258]]	4.56
LogisticRegression	0.87	0.87	0.87	0.87	[[2244 250] [412 2309]]	0.29
NearestCentroid	0.72	0.73	0.73	0.71	[[2279 215] [1236 1485]]	0.11
NuSVC	0.86	0.86	0.86	0.86	[[2172 322] [414 2307]]	55.69
PassiveAggressiveClassifier	0.80	0.80	0.80	0.80	[[1949 545] [501 2220]]	0.17
Perceptron	0.79	0.79	0.79	0.79	[[2095 399] [693 2028]]	0.16
QuadraticDiscriminantAnalysis	0.54	0.52	0.52	0.46	[[340 2154] [238 2483]]	0.26
RandomForestClassifier	0.92	0.92	0.92	0.92	[[2144 350] [60 2661]]	3.86
RidgeClassifier	0.81	0.81	0.81	0.81	[[2090 404] [599 2122]]	0.15
RidgeClassifierCV	0.81	0.81	0.81	0.81	[[2090 404] [599 2122]]	0.31
SGDClassifier	0.87	0.87	0.87	0.87	[[2220 274] [415 2306]]	0.38
SVC	0.90	0.90	0.90	0.90	[[2192 302] [229 2492]]	26.51
XGBClassifier	0.93	0.92	0.92	0.93	[[2210 284] [104 2617]]	3.33

In order to show the predictions over all 40 weeks, using the format in Table 4.16, it would have required displaying 40 different tables. Thus, all the results for each week were stored separately. The individual metrics were then extracted and stored together for each of the 40 weeks. These metrics as well as the confusion matrix will be discussed next.

4.6.2.1 Confusion Matrix

A confusion matrix summarizes the performance of a ML model in a tabular form, in an NxN matrix. Since the SAR problem is a binary classification problem, the confusion matrix is a 2x2 matrix. The confusion matrix for all 25 MLAs, for each of the 40 weeks, is shown in Table A1 in the appendix.

The confusion matrix is not a performance measure, but it can be used to calculate other metrics. As an example, the confusion matrix for the XGBClassifier for Week 39, is shown in Table 4.17.

Table 4.17: Confusion matrix for XGBClassifier for Week 39. Source: Researcher's own creation.

n = 5215		**Actual**	
		At Risk	**Not At Risk**
Predicted	**At Risk**	2165	329
	Not At Risk	104	2617

Since there was a total of 26 071 samples, and the training versus testing split was 80:20, the total number of samples for the prediction was 5215. The accuracy, for instance, was 0.92 or 92%. The accuracy metric will be discussed further in the next section.

4.6.2.2 Accuracy

Accuracy refers to the ratio of correct predictions to the total number of predictions. In essence, of all the predictions that were made for identifying SAR, accuracy asks the question: "What ratio of students was correctly identified as either being at risk, or being not at risk?" From the

values in the confusion matrix, shown in Table 4.17, the accuracy for the XGBClassifier for week 39 can be calculated as follows:

$$Accuracy = \frac{2165 + 2617}{2165 + 329 + 104 + 2617} = 0.92 \text{ (or } 92\%)$$

The accuracy scores (rounded to 2 decimal places), for all 25 MLAs for each of the 40 weeks, is shown in Table 4.18. The ML algorithm(s) with the highest accuracy for each week is highlighted in green. From Table 4.18, it can be observed that the LGBMClassifier algorithm has the highest accuracy for 29 out of the 40 weeks. The RandomForestClassifier algorithm has the highest accuracy for 8 out of the 40 weeks, while the AdaBoostClassifier has the highest accuracy for 3 out of the 40 weeks.

Table 4.18: Accuracy scores for 25 machine learning algorithms over 40 weeks.

ML Algorithm	Wk0	Wk1	Wk2	Wk3	Wk4	Wk5	Wk6	Wk7	Wk8	Wk9	Wk10	Wk11	Wk12	Wk13	Wk14	Wk15	Wk16	Wk17	Wk18	Wk19	Wk20	Wk21	Wk22	Wk23	Wk24	Wk25	Wk26	Wk27	Wk28	Wk29	Wk30	Wk31	Wk32	Wk33	Wk34	Wk35	Wk36	Wk37	Wk38	Wk39
AdaBoostClassifier	0.64	0.66	0.67	0.68	0.69	0.69	0.71	0.73	0.73	0.74	0.75	0.75	0.75	0.77	0.77	0.78	0.79	0.79	0.81	0.81	0.82	0.82	0.83	0.84	0.85	0.85	0.86	0.86	0.86	0.87	0.87	0.88	0.88	0.89	0.90	0.89	0.90	0.90	0.90	0.90
BaggingClassifier	0.59	0.61	0.63	0.63	0.65	0.66	0.68	0.69	0.71	0.70	0.71	0.71	0.71	0.74	0.74	0.76	0.76	0.78	0.79	0.80	0.80	0.82	0.82	0.84	0.85	0.85	0.85	0.86	0.87	0.87	0.88	0.89	0.89	0.90	0.91	0.91	0.91	0.90	0.90	0.91
BernoulliNB	0.62	0.63	0.65	0.65	0.64	0.64	0.64	0.65	0.65	0.66	0.66	0.66	0.67	0.67	0.68	0.68	0.69	0.70	0.71	0.71	0.72	0.72	0.73	0.73	0.74	0.74	0.75	0.75	0.75	0.76	0.77	0.77	0.77	0.78	0.78	0.79	0.79	0.79	0.79	0.79
CalibratedClassifierCV	0.63	0.64	0.65	0.66	0.66	0.67	0.68	0.69	0.69	0.70	0.70	0.70	0.71	0.71	0.72	0.73	0.74	0.74	0.74	0.75	0.76	0.77	0.78	0.79	0.79	0.80	0.80	0.81	0.82	0.83	0.83	0.83	0.84	0.84	0.85	0.85	0.86	0.86	0.86	0.85
DecisionTreeClassifier	0.58	0.57	0.59	0.60	0.60	0.62	0.64	0.65	0.65	0.64	0.68	0.67	0.68	0.68	0.70	0.71	0.72	0.72	0.73	0.75	0.75	0.76	0.77	0.79	0.79	0.80	0.81	0.82	0.82	0.83	0.83	0.83	0.85	0.85	0.86	0.86	0.86	0.86	0.86	0.86
DummyClassifier	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
ExtraTreeClassifier	0.57	0.56	0.57	0.58	0.59	0.60	0.61	0.62	0.62	0.64	0.65	0.65	0.65	0.66	0.67	0.69	0.69	0.70	0.70	0.73	0.72	0.75	0.75	0.76	0.76	0.77	0.77	0.78	0.78	0.79	0.79	0.80	0.81	0.81	0.83	0.83	0.83	0.84	0.82	0.82
ExtraTreesClassifier	0.58	0.61	0.63	0.65	0.66	0.68	0.70	0.71	0.72	0.73	0.74	0.74	0.75	0.76	0.77	0.79	0.79	0.80	0.81	0.81	0.82	0.83	0.84	0.85	0.86	0.87	0.87	0.87	0.88	0.89	0.89	0.90	0.90	0.90	0.91	0.91	0.91	0.91	0.91	0.91
GaussianNB	0.59	0.59	0.60	0.60	0.60	0.60	0.60	0.61	0.60	0.60	0.59	0.59	0.59	0.60	0.60	0.61	0.61	0.62	0.63	0.63	0.64	0.64	0.65	0.66	0.66	0.66	0.67	0.67	0.68	0.68	0.68	0.68	0.68	0.69	0.69	0.69	0.69	0.69	0.69	0.69
KNeighborsClassifier	0.58	0.60	0.61	0.61	0.62	0.63	0.64	0.65	0.65	0.65	0.66	0.66	0.66	0.67	0.67	0.68	0.68	0.69	0.69	0.70	0.71	0.71	0.72	0.72	0.73	0.73	0.73	0.73	0.74	0.74	0.75	0.75	0.76	0.76	0.76	0.76	0.77	0.76	0.76	0.76
LGBMClassifier	0.64	0.66	0.67	0.67	0.69	0.70	0.72	0.74	0.74	0.75	0.76	0.76	0.76	0.77	0.78	0.79	0.80	0.81	0.82	0.82	0.84	0.84	0.86	0.86	0.87	0.88	0.88	0.88	0.88	0.89	0.90	0.91	0.91	0.91	0.92	0.92	0.92	0.92	0.92	0.92
LinearDiscriminantAnalysis	0.62	0.63	0.65	0.65	0.66	0.67	0.67	0.68	0.68	0.68	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.73	0.74	0.74	0.75	0.75	0.75	0.75	0.76	0.76	0.77	0.77	0.77	0.77	0.78	0.79	0.79	0.80	0.80	0.80	0.80
LinearSVC	0.63	0.64	0.65	0.66	0.66	0.67	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.73	0.74	0.74	0.76	0.76	0.77	0.78	0.79	0.79	0.80	0.80	0.81	0.81	0.82	0.83	0.83	0.84	0.84	0.84	0.85	0.85	0.85	0.85	0.85
LogisticRegression	0.63	0.64	0.66	0.66	0.67	0.68	0.68	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.74	0.74	0.75	0.75	0.77	0.77	0.78	0.79	0.80	0.81	0.81	0.81	0.82	0.83	0.84	0.85	0.85	0.85	0.85	0.86	0.87	0.86	0.87	0.87	0.86
NearestCentroid	0.62	0.63	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.65	0.64	0.64	0.64	0.65	0.65	0.65	0.66	0.66	0.66	0.67	0.67	0.68	0.68	0.68	0.69	0.69	0.69	0.69	0.69	0.70	0.70	0.70	0.70	0.71	0.71	0.71	0.71	0.71	0.71	0.71
NuSVC	0.59	0.60	0.62	0.63	0.64	0.65	0.66	0.68	0.69	0.70	0.72	0.72	0.72	0.73	0.76	0.77	0.78	0.78	0.79	0.80	0.80	0.80	0.81	0.82	0.82	0.83	0.83	0.83	0.83	0.84	0.84	0.84	0.84	0.85	0.85	0.85	0.85	0.85	0.85	0.85
PassiveAggressiveClassifier	0.55	0.55	0.59	0.60	0.61	0.61	0.58	0.61	0.62	0.63	0.63	0.64	0.64	0.65	0.67	0.62	0.63	0.62	0.63	0.63	0.71	0.74	0.75	0.68	0.69	0.68	0.73	0.68	0.79	0.80	0.78	0.81	0.79	0.73	0.74	0.75	0.75	0.75	0.75	0.75
Perceptron	0.52	0.53	0.55	0.60	0.58	0.61	0.59	0.64	0.60	0.62	0.63	0.64	0.61	0.65	0.68	0.65	0.70	0.70	0.71	0.71	0.74	0.75	0.74	0.71	0.77	0.78	0.78	0.75	0.78	0.80	0.78	0.78	0.73	0.80	0.80	0.82	0.80	0.81	0.80	0.80
QuadraticDiscriminantAnalysis	0.58	0.56	0.59	0.57	0.56	0.57	0.60	0.57	0.58	0.59	0.60	0.60	0.60	0.63	0.63	0.63	0.63	0.63	0.64	0.65	0.66	0.65	0.66	0.67	0.66	0.68	0.67	0.68	0.68	0.70	0.69	0.70	0.68	0.69	0.70	0.69	0.70	0.70	0.71	0.70
RandomForestClassifier	0.61	0.64	0.66	0.67	0.68	0.69	0.71	0.72	0.73	0.74	0.75	0.75	0.76	0.77	0.77	0.79	0.79	0.81	0.81	0.83	0.83	0.83	0.85	0.86	0.86	0.87	0.88	0.88	0.89	0.89	0.90	0.90	0.91	0.91	0.92	0.92	0.92	0.92	0.92	0.92
RidgeClassifier	0.62	0.63	0.65	0.65	0.66	0.67	0.67	0.68	0.68	0.68	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.73	0.74	0.74	0.75	0.75	0.75	0.75	0.75	0.76	0.77	0.77	0.77	0.77	0.78	0.79	0.79	0.79	0.80	0.80	0.80
RidgeClassifierCV	0.62	0.63	0.65	0.65	0.66	0.67	0.67	0.68	0.68	0.68	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.73	0.74	0.74	0.75	0.75	0.75	0.75	0.75	0.76	0.77	0.77	0.77	0.77	0.78	0.79	0.79	0.80	0.80	0.80	0.80
SGDClassifier	0.60	0.62	0.62	0.63	0.64	0.66	0.66	0.68	0.67	0.68	0.69	0.69	0.69	0.70	0.71	0.73	0.72	0.72	0.75	0.76	0.76	0.77	0.79	0.79	0.79	0.80	0.81	0.82	0.81	0.82	0.83	0.83	0.84	0.84	0.85	0.86	0.86	0.86	0.87	0.86
SVC	0.63	0.65	0.66	0.66	0.67	0.68	0.70	0.71	0.71	0.72	0.73	0.73	0.74	0.75	0.76	0.77	0.78	0.78	0.79	0.80	0.81	0.81	0.82	0.83	0.84	0.85	0.85	0.86	0.86	0.87	0.87	0.88	0.88	0.89	0.89	0.89	0.89	0.89	0.89	0.89
XGBClassifier	0.63	0.64	0.66	0.66	0.68	0.69	0.71	0.72	0.73	0.73	0.74	0.75	0.75	0.76	0.77	0.78	0.79	0.80	0.81	0.82	0.83	0.84	0.84	0.85	0.86	0.87	0.88	0.87	0.88	0.89	0.90	0.90	0.90	0.91	0.92	0.92	0.92	0.92	0.92	0.92

4.6.2.3 Balanced Accuracy

Balanced accuracy is another performance metric that can be used to measure the performance of a ML classifier. It is the arithmetic mean of two other metrics called sensitivity (also known as the true positive rate or recall) and specificity (also known as the true negative rate).

Sensitivity asks the question: "Out of all the students that were actually at risk, what ratio of students was correctly identified as being at risk?" Specificity asks the question: "Out of all the students that were actually not at risk, what ratio of students was correctly identified as being not at risk?"

From the values in the confusion matrix, displayed in Table 4.17, the balanced accuracy for the XGBClassifier for week 39 can be calculated as follows:

$$Sensitivity\ (Recall) = \frac{2165}{2165 + 104} = 0.95\ (\text{or } 95\%)$$

$$Specificity = \frac{2617}{329 + 2617} = 0.89\ (\text{or } 89\%)$$

$$Balanced\ Accuracy = \frac{0.95 + 0.89}{2} = 0.92\ (\text{or } 92\%)$$

The balanced accuracy scores (rounded to 2 decimal places), for all 25 MLAs for each of the 40 weeks, is shown in Table 4.19. The ML algorithm(s) with the highest accuracy for each week is highlighted in green. From Table 4.19, it can be observed that the LGBMClassifier algorithm has the highest balanced accuracy for 33 out of the 40 weeks. The AdaBoostClassifier has the highest accuracy for 3 out of the 40 weeks. Both the RandomForestClassifier and XGBClassifier algorithms have the highest accuracy for 2 out of the 40 weeks.

Given that the dataset used was nearly balanced, the balanced accuracy scores did not differ by much from the accuracy scores, as can be seen from Table 4.18 and Table 4.19.

Table 4.19: Balanced accuracy scores for 25 machine learning algorithms over 40 weeks.

ML Algorithm	Wk0	Wk1	Wk2	Wk3	Wk4	Wk5	Wk6	Wk7	Wk8	Wk9	Wk10	Wk11	Wk12	Wk13	Wk14	Wk15	Wk16	Wk17	Wk18	Wk19	Wk20	Wk21	Wk22	Wk23	Wk24	Wk25	Wk26	Wk27	Wk28	Wk29	Wk30	Wk31	Wk32	Wk33	Wk34	Wk35	Wk36	Wk37	Wk38	Wk39
AdaBoostClassifier	0.64	0.66	0.66	0.67	0.68	0.69	0.71	0.73	0.73	0.74	0.74	0.75	0.75	0.77	0.77	0.78	0.79	0.79	0.81	0.81	0.82	0.82	0.83	0.84	0.85	0.85	0.86	0.86	0.86	0.87	0.87	0.88	0.88	0.89	0.90	0.89	0.90	0.90	0.90	0.90
BaggingClassifier	0.59	0.61	0.63	0.63	0.65	0.66	0.68	0.69	0.71	0.70	0.71	0.71	0.71	0.73	0.74	0.76	0.76	0.78	0.79	0.80	0.80	0.82	0.82	0.84	0.84	0.85	0.85	0.85	0.87	0.87	0.88	0.88	0.89	0.89	0.90	0.91	0.90	0.90	0.90	0.90
BernoulliNB	0.62	0.63	0.65	0.65	0.64	0.64	0.65	0.65	0.65	0.66	0.67	0.67	0.67	0.68	0.68	0.68	0.69	0.70	0.71	0.71	0.72	0.73	0.73	0.74	0.74	0.75	0.75	0.75	0.76	0.77	0.77	0.78	0.78	0.78	0.79	0.79	0.79	0.80	0.80	0.80
CalibratedClassifierCV	0.62	0.64	0.65	0.66	0.66	0.67	0.68	0.69	0.69	0.70	0.70	0.70	0.71	0.71	0.72	0.73	0.74	0.74	0.75	0.75	0.76	0.77	0.78	0.79	0.79	0.80	0.80	0.81	0.82	0.83	0.83	0.83	0.84	0.84	0.85	0.86	0.86	0.86	0.86	0.86
DecisionTreeClassifier	0.58	0.57	0.59	0.60	0.60	0.62	0.64	0.65	0.65	0.64	0.68	0.67	0.68	0.68	0.70	0.71	0.72	0.72	0.74	0.75	0.75	0.76	0.77	0.79	0.79	0.80	0.81	0.82	0.82	0.83	0.83	0.83	0.85	0.85	0.86	0.86	0.86	0.86	0.86	0.86
DummyClassifier	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
ExtraTreeClassifier	0.57	0.56	0.57	0.58	0.59	0.60	0.61	0.62	0.62	0.64	0.64	0.65	0.65	0.66	0.67	0.69	0.69	0.70	0.70	0.73	0.72	0.74	0.75	0.76	0.76	0.77	0.77	0.78	0.78	0.79	0.79	0.80	0.81	0.81	0.83	0.83	0.83	0.84	0.82	0.82
ExtraTreesClassifier	0.58	0.61	0.63	0.65	0.66	0.68	0.69	0.71	0.71	0.72	0.74	0.74	0.75	0.76	0.77	0.78	0.78	0.79	0.81	0.81	0.82	0.83	0.84	0.85	0.85	0.86	0.86	0.87	0.87	0.88	0.89	0.89	0.90	0.90	0.91	0.91	0.91	0.91	0.91	0.91
GaussianNB	0.59	0.60	0.60	0.61	0.61	0.61	0.61	0.62	0.61	0.61	0.60	0.60	0.60	0.61	0.62	0.62	0.63	0.63	0.64	0.64	0.65	0.65	0.66	0.66	0.67	0.67	0.68	0.68	0.69	0.69	0.69	0.69	0.69	0.70	0.70	0.70	0.70	0.70	0.70	0.70
KNeighborsClassifier	0.58	0.60	0.61	0.61	0.62	0.63	0.64	0.65	0.65	0.65	0.66	0.66	0.66	0.67	0.67	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.72	0.73	0.73	0.73	0.74	0.74	0.74	0.75	0.75	0.76	0.76	0.76	0.77	0.77	0.77	0.77	0.77	0.77
LGBMClassifier	0.64	0.65	0.67	0.67	0.69	0.70	0.72	0.73	0.73	0.74	0.75	0.76	0.76	0.77	0.78	0.79	0.80	0.81	0.81	0.82	0.83	0.84	0.85	0.85	0.86	0.87	0.87	0.87	0.88	0.89	0.90	0.90	0.91	0.91	0.91	0.92	0.92	0.92	0.92	0.92
LinearDiscriminantAnalysis	0.62	0.63	0.65	0.65	0.66	0.67	0.67	0.68	0.68	0.68	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.73	0.74	0.75	0.75	0.75	0.75	0.75	0.76	0.76	0.77	0.77	0.77	0.78	0.78	0.79	0.80	0.80	0.80	0.80	0.80
LinearSVC	0.63	0.64	0.65	0.65	0.66	0.67	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.74	0.74	0.74	0.76	0.76	0.77	0.78	0.79	0.79	0.80	0.80	0.81	0.82	0.82	0.83	0.84	0.84	0.84	0.84	0.86	0.86	0.86	0.86	0.86
LogisticRegression	0.63	0.64	0.66	0.66	0.67	0.67	0.68	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.74	0.74	0.75	0.75	0.77	0.77	0.78	0.79	0.80	0.81	0.82	0.82	0.82	0.83	0.84	0.85	0.85	0.85	0.85	0.86	0.87	0.87	0.87	0.87	0.87
NearestCentroid	0.62	0.63	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.66	0.66	0.66	0.67	0.67	0.67	0.68	0.68	0.68	0.69	0.69	0.70	0.70	0.70	0.70	0.70	0.71	0.71	0.71	0.71	0.72	0.72	0.72	0.72	0.72	0.72	0.72
NuSVC	0.59	0.60	0.62	0.63	0.64	0.65	0.66	0.68	0.69	0.70	0.71	0.72	0.72	0.73	0.75	0.77	0.78	0.78	0.79	0.80	0.80	0.80	0.81	0.82	0.82	0.83	0.83	0.83	0.83	0.84	0.84	0.84	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
PassiveAggressiveClassifier	0.55	0.55	0.59	0.60	0.60	0.60	0.58	0.61	0.62	0.63	0.63	0.63	0.64	0.65	0.67	0.62	0.62	0.62	0.62	0.62	0.71	0.75	0.76	0.68	0.69	0.68	0.73	0.68	0.79	0.80	0.78	0.81	0.79	0.73	0.73	0.74	0.74	0.74	0.75	0.75
Perceptron	0.52	0.53	0.55	0.61	0.58	0.61	0.59	0.64	0.60	0.62	0.63	0.64	0.61	0.65	0.68	0.65	0.70	0.70	0.71	0.72	0.74	0.75	0.75	0.71	0.77	0.78	0.78	0.75	0.78	0.80	0.78	0.78	0.73	0.80	0.80	0.82	0.80	0.82	0.80	0.80
QuadraticDiscriminantAnalysis	0.58	0.56	0.59	0.57	0.55	0.58	0.60	0.58	0.59	0.60	0.60	0.60	0.61	0.64	0.64	0.64	0.63	0.64	0.65	0.65	0.67	0.66	0.67	0.68	0.67	0.69	0.68	0.68	0.69	0.70	0.70	0.70	0.69	0.70	0.71	0.70	0.71	0.71	0.71	0.71
RandomForestClassifier	0.61	0.64	0.65	0.67	0.68	0.69	0.71	0.72	0.73	0.74	0.74	0.75	0.75	0.76	0.77	0.79	0.79	0.80	0.81	0.82	0.83	0.83	0.85	0.85	0.86	0.87	0.87	0.88	0.88	0.89	0.89	0.90	0.91	0.91	0.91	0.91	0.91	0.92	0.91	0.91
RidgeClassifier	0.62	0.63	0.65	0.65	0.66	0.67	0.67	0.68	0.68	0.68	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.72	0.72	0.73	0.73	0.74	0.75	0.75	0.75	0.75	0.75	0.76	0.76	0.77	0.77	0.77	0.78	0.78	0.79	0.80	0.80	0.80	0.80	0.80
RidgeClassifierCV	0.62	0.63	0.65	0.65	0.66	0.67	0.67	0.68	0.68	0.68	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.73	0.74	0.75	0.75	0.75	0.75	0.75	0.76	0.76	0.77	0.77	0.77	0.78	0.78	0.79	0.80	0.80	0.80	0.80	0.80
SGDClassifier	0.60	0.62	0.63	0.63	0.63	0.65	0.67	0.68	0.67	0.68	0.69	0.69	0.69	0.70	0.70	0.73	0.72	0.72	0.75	0.76	0.76	0.77	0.79	0.79	0.80	0.80	0.81	0.81	0.81	0.82	0.83	0.84	0.85	0.85	0.85	0.86	0.86	0.86	0.87	0.86
SVC	0.63	0.64	0.66	0.66	0.67	0.68	0.70	0.71	0.71	0.72	0.72	0.73	0.74	0.74	0.76	0.77	0.77	0.78	0.79	0.80	0.81	0.81	0.82	0.83	0.84	0.85	0.85	0.85	0.86	0.86	0.87	0.88	0.88	0.89	0.89	0.89	0.89	0.89	0.89	0.89
XGBClassifier	0.63	0.64	0.65	0.66	0.67	0.69	0.71	0.71	0.72	0.73	0.74	0.75	0.75	0.76	0.77	0.78	0.79	0.80	0.81	0.82	0.82	0.83	0.84	0.85	0.86	0.87	0.87	0.87	0.88	0.88	0.90	0.89	0.90	0.91	0.91	0.92	0.91	0.92	0.91	0.91

4.6.2.4 F1 Scores

The F1 score is the harmonic mean, or weighted average, of two other metrics called precision and recall (or sensitivity). Recall asks the question: "Out of all the students that were actually at risk, what ratio of students was correctly identified as being at risk?" Precision asks the question: "Of all the students predicted as being at risk, what ratio of students was correctly predicted as being at risk?"

From the values in the confusion matrix, depicted in Table 4.17, the F1 score for the XGBClassifier for week 39, can be calculated as follows:

$$Recall = \frac{2165}{2165 + 104} = 0.95 \text{ (or } 95\% \text{)}$$

$$Precision = \frac{2165}{2165 + 329} = 0.87 \text{ (or } 87\%)$$

$$F1\ Score = 2 * \frac{(0.87 * 0.95)}{0.87 + 0.95} = 0.91 \text{ (or } 91\%)$$

The F1 scores (rounded to 2 decimal places), for the all 25 MLAs for each of the 40 weeks, is shown in Table 4.20. Due to rounding off, the F1 score for the example above differs slightly from the score shown in Table 4.20. The ML algorithm(s) with the highest accuracy for each week is highlighted in green. From Table 4.20, it can be observed that the LGBMClassifier algorithm has the highest F1 score for 31 out of the 40 weeks. The RandomForestClassifier algorithm has the highest accuracy for 4 out of the 40 weeks. The AdaBoostClassifier has the highest accuracy for 3 out of the 40 weeks. The XGBClassifier algorithm has the highest accuracy for 2 out of the 40 weeks.

Table 4.20: F1 Scores for 25 machine learning algorithms over 40 weeks.

ML Algorithm	Wk0	Wk1	Wk2	Wk3	Wk4	Wk5	Wk6	Wk7	Wk8	Wk9	Wk10	Wk11	Wk12	Wk13	Wk14	Wk15	Wk16	Wk17	Wk18	Wk19	Wk20	Wk21	Wk22	Wk23	Wk24	Wk25	Wk26	Wk27	Wk28	Wk29	Wk30	Wk31	Wk32	Wk33	Wk34	Wk35	Wk36	Wk37	Wk38	Wk39
AdaBoostClassifier	0.64	0.66	0.67	0.68	0.68	0.69	0.71	0.73	0.73	0.74	0.75	0.75	0.75	0.77	0.77	0.78	0.79	0.79	0.81	0.81	0.82	0.82	0.83	0.84	0.85	0.85	0.86	0.86	0.86	0.87	0.87	0.88	0.88	0.89	0.90	0.89	0.90	0.90	0.90	0.90
BaggingClassifier	0.59	0.61	0.63	0.63	0.65	0.66	0.68	0.69	0.71	0.70	0.71	0.71	0.71	0.74	0.74	0.76	0.76	0.78	0.79	0.80	0.80	0.82	0.82	0.84	0.84	0.85	0.85	0.86	0.87	0.87	0.88	0.89	0.89	0.90	0.91	0.91	0.90	0.90	0.90	0.91
BernoulliNB	0.62	0.63	0.65	0.65	0.64	0.64	0.64	0.65	0.65	0.65	0.66	0.66	0.66	0.67	0.67	0.67	0.68	0.69	0.70	0.71	0.71	0.72	0.72	0.73	0.74	0.74	0.74	0.75	0.75	0.76	0.77	0.77	0.77	0.78	0.78	0.79	0.79	0.79	0.79	0.79
CalibratedClassifierCV	0.63	0.64	0.65	0.66	0.66	0.67	0.68	0.69	0.69	0.70	0.70	0.70	0.71	0.71	0.72	0.73	0.74	0.74	0.74	0.75	0.76	0.77	0.78	0.79	0.79	0.80	0.80	0.81	0.82	0.83	0.83	0.83	0.84	0.84	0.85	0.85	0.86	0.86	0.85	0.85
DecisionTreeClassifier	0.58	0.57	0.59	0.60	0.60	0.62	0.64	0.65	0.65	0.64	0.68	0.67	0.68	0.68	0.70	0.71	0.72	0.72	0.74	0.75	0.75	0.76	0.77	0.79	0.79	0.80	0.81	0.82	0.82	0.83	0.83	0.83	0.85	0.85	0.86	0.86	0.86	0.86	0.86	0.86
DummyClassifier	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
ExtraTreeClassifier	0.57	0.56	0.57	0.58	0.59	0.60	0.61	0.62	0.62	0.64	0.65	0.65	0.65	0.66	0.67	0.69	0.69	0.70	0.70	0.73	0.72	0.75	0.75	0.76	0.76	0.77	0.77	0.78	0.78	0.79	0.79	0.80	0.81	0.81	0.83	0.83	0.83	0.84	0.82	0.82
ExtraTreesClassifier	0.58	0.61	0.63	0.65	0.66	0.68	0.70	0.71	0.72	0.73	0.74	0.74	0.75	0.76	0.77	0.78	0.78	0.79	0.81	0.81	0.82	0.83	0.84	0.85	0.86	0.87	0.87	0.87	0.88	0.88	0.89	0.89	0.90	0.90	0.91	0.91	0.91	0.91	0.91	0.91
GaussianNB	0.58	0.59	0.59	0.59	0.59	0.58	0.59	0.58	0.57	0.57	0.56	0.56	0.56	0.57	0.58	0.58	0.59	0.60	0.61	0.61	0.62	0.62	0.63	0.64	0.64	0.65	0.65	0.66	0.66	0.66	0.67	0.67	0.67	0.67	0.68	0.68	0.67	0.68	0.68	0.68
KNeighborsClassifier	0.58	0.60	0.61	0.61	0.62	0.63	0.64	0.65	0.65	0.65	0.66	0.66	0.66	0.67	0.67	0.68	0.68	0.69	0.69	0.70	0.71	0.71	0.72	0.72	0.73	0.73	0.73	0.73	0.74	0.74	0.75	0.75	0.75	0.76	0.76	0.76	0.76	0.76	0.76	0.76
LGBMClassifier	0.64	0.65	0.67	0.67	0.69	0.70	0.72	0.74	0.74	0.75	0.75	0.76	0.76	0.77	0.78	0.79	0.80	0.81	0.82	0.82	0.83	0.84	0.85	0.86	0.87	0.87	0.87	0.88	0.88	0.89	0.90	0.90	0.91	0.91	0.92	0.92	0.92	0.92	0.92	0.92
LinearDiscriminantAnalysis	0.62	0.63	0.65	0.65	0.66	0.67	0.67	0.68	0.68	0.68	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.73	0.74	0.74	0.75	0.75	0.75	0.75	0.76	0.76	0.77	0.77	0.77	0.77	0.78	0.79	0.79	0.80	0.80	0.80	0.80
LinearSVC	0.63	0.64	0.65	0.65	0.66	0.67	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.73	0.74	0.74	0.76	0.76	0.77	0.78	0.79	0.79	0.80	0.80	0.81	0.81	0.82	0.83	0.83	0.84	0.84	0.84	0.85	0.85	0.85	0.85	0.85
LogisticRegression	0.63	0.64	0.66	0.66	0.67	0.68	0.68	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.74	0.74	0.75	0.75	0.77	0.77	0.78	0.79	0.80	0.81	0.81	0.81	0.82	0.83	0.84	0.85	0.85	0.85	0.85	0.86	0.87	0.86	0.87	0.87	0.86
NearestCentroid	0.62	0.63	0.64	0.64	0.64	0.64	0.64	0.63	0.63	0.64	0.63	0.63	0.63	0.64	0.64	0.64	0.65	0.65	0.65	0.66	0.66	0.67	0.67	0.67	0.68	0.68	0.68	0.68	0.68	0.68	0.69	0.69	0.69	0.70	0.70	0.70	0.70	0.70	0.70	0.70
NuSVC	0.59	0.60	0.62	0.63	0.64	0.65	0.66	0.68	0.69	0.70	0.71	0.72	0.72	0.73	0.76	0.77	0.78	0.78	0.79	0.80	0.80	0.80	0.81	0.82	0.82	0.83	0.83	0.83	0.83	0.84	0.84	0.84	0.84	0.85	0.85	0.85	0.85	0.85	0.85	0.85
PassiveAggressiveClassifier	0.55	0.55	0.59	0.60	0.60	0.60	0.58	0.61	0.62	0.63	0.63	0.64	0.64	0.65	0.67	0.62	0.62	0.62	0.63	0.63	0.71	0.74	0.75	0.68	0.69	0.68	0.73	0.68	0.79	0.80	0.78	0.81	0.79	0.73	0.74	0.75	0.75	0.75	0.75	0.75
Perceptron	0.52	0.53	0.55	0.60	0.58	0.61	0.59	0.64	0.60	0.62	0.63	0.64	0.61	0.65	0.68	0.65	0.70	0.70	0.71	0.71	0.74	0.75	0.74	0.70	0.77	0.78	0.78	0.75	0.78	0.80	0.78	0.78	0.73	0.80	0.80	0.82	0.80	0.81	0.80	0.80
QuadraticDiscriminantAnalysis	0.58	0.55	0.58	0.57	0.55	0.57	0.59	0.55	0.57	0.57	0.59	0.59	0.60	0.63	0.62	0.61	0.62	0.61	0.63	0.63	0.65	0.63	0.65	0.65	0.65	0.66	0.66	0.67	0.67	0.69	0.67	0.68	0.66	0.68	0.69	0.67	0.69	0.69	0.69	0.70
RandomForestClassifier	0.61	0.64	0.66	0.67	0.68	0.69	0.71	0.72	0.73	0.74	0.74	0.75	0.76	0.76	0.77	0.79	0.79	0.80	0.81	0.82	0.83	0.83	0.85	0.86	0.86	0.87	0.88	0.88	0.89	0.89	0.90	0.90	0.91	0.91	0.92	0.92	0.92	0.92	0.92	0.92
RidgeClassifier	0.62	0.63	0.65	0.65	0.66	0.67	0.67	0.68	0.68	0.68	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.73	0.74	0.74	0.75	0.75	0.75	0.75	0.75	0.76	0.77	0.77	0.77	0.77	0.78	0.79	0.79	0.79	0.80	0.80	0.80
RidgeClassifierCV	0.62	0.63	0.65	0.65	0.66	0.67	0.67	0.68	0.68	0.68	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.73	0.74	0.74	0.75	0.75	0.75	0.75	0.75	0.76	0.77	0.77	0.77	0.77	0.78	0.79	0.79	0.80	0.80	0.80	0.80
SGDClassifier	0.60	0.62	0.62	0.63	0.64	0.66	0.66	0.68	0.67	0.68	0.69	0.69	0.69	0.70	0.70	0.73	0.72	0.72	0.75	0.76	0.75	0.77	0.79	0.79	0.79	0.80	0.81	0.82	0.81	0.81	0.82	0.83	0.84	0.84	0.85	0.86	0.86	0.86	0.87	0.86
SVC	0.63	0.64	0.66	0.66	0.67	0.68	0.70	0.71	0.71	0.72	0.72	0.73	0.74	0.75	0.76	0.77	0.78	0.78	0.79	0.80	0.81	0.81	0.82	0.83	0.84	0.85	0.85	0.86	0.86	0.87	0.87	0.88	0.88	0.89	0.89	0.89	0.89	0.89	0.89	0.89
XGBClassifier	0.63	0.64	0.65	0.66	0.67	0.69	0.71	0.72	0.73	0.73	0.74	0.75	0.75	0.76	0.77	0.78	0.79	0.80	0.81	0.82	0.83	0.84	0.84	0.85	0.86	0.87	0.88	0.87	0.88	0.88	0.90	0.90	0.90	0.91	0.92	0.92	0.92	0.92	0.92	0.92

4.6.2.5 ROC AUC Scores

The ROC AUC score gives the probability that a randomly chosen positive instance (student is at risk), is higher than a randomly chosen negative instance (student is not at risk).

The ROC AUC scores (rounded to 2 decimal places) for all 25 MLAs for each of the 40 weeks is shown in Table 4.21. The ML algorithm(s) with the highest accuracy for each week is highlighted in green. From Table 4.21, it can be observed that the LGBMClassifier algorithm has the highest ROC AUC score for 33 out of the 40 weeks. The AdaBoostClassifier has the highest accuracy for 3 out of the 40 weeks. Both the RandomForestClassifier and XGBClassifier algorithms have the highest accuracy for 2 out of the 40 weeks.

Table 4.21: ROC AUC scores for 25 machine learning algorithms over 40 weeks.

ML Algorithm	Wk0	Wk1	Wk2	Wk3	Wk4	Wk5	Wk6	Wk7	Wk8	Wk9	Wk10	Wk11	Wk12	Wk13	Wk14	Wk15	Wk16	Wk17	Wk18	Wk19	Wk20	Wk21	Wk22	Wk23	Wk24	Wk25	Wk26	Wk27	Wk28	Wk29	Wk30	Wk31	Wk32	Wk33	Wk34	Wk35	Wk36	Wk37	Wk38	Wk39
AdaBoostClassifier	0.64	0.66	0.66	0.67	0.68	0.69	0.71	0.73	0.73	0.74	0.74	0.75	0.75	0.77	0.77	0.78	0.79	0.79	0.81	0.81	0.82	0.82	0.83	0.84	0.85	0.85	0.86	0.86	0.86	0.87	0.87	0.88	0.88	0.89	0.90	0.89	0.90	0.90	0.90	0.90
BaggingClassifier	0.59	0.61	0.63	0.63	0.65	0.66	0.68	0.69	0.71	0.70	0.71	0.71	0.71	0.73	0.74	0.76	0.76	0.78	0.79	0.80	0.80	0.82	0.82	0.84	0.84	0.85	0.85	0.85	0.87	0.87	0.88	0.88	0.89	0.89	0.90	0.91	0.90	0.90	0.90	0.90
BernoulliNB	0.62	0.63	0.65	0.65	0.64	0.64	0.65	0.65	0.65	0.66	0.67	0.67	0.67	0.68	0.68	0.68	0.69	0.70	0.71	0.71	0.72	0.73	0.73	0.74	0.74	0.75	0.75	0.75	0.76	0.77	0.77	0.78	0.78	0.78	0.79	0.79	0.79	0.80	0.80	0.80
CalibratedClassifierCV	0.62	0.64	0.65	0.66	0.66	0.67	0.68	0.69	0.69	0.70	0.70	0.70	0.71	0.71	0.72	0.73	0.74	0.74	0.75	0.75	0.76	0.77	0.78	0.79	0.79	0.80	0.80	0.81	0.82	0.83	0.83	0.83	0.84	0.84	0.85	0.86	0.86	0.86	0.86	0.86
DecisionTreeClassifier	0.58	0.57	0.59	0.60	0.60	0.62	0.64	0.65	0.65	0.64	0.68	0.67	0.68	0.68	0.70	0.71	0.72	0.72	0.74	0.75	0.75	0.76	0.77	0.79	0.79	0.80	0.81	0.82	0.82	0.83	0.83	0.83	0.85	0.85	0.86	0.86	0.86	0.86	0.86	0.86
DummyClassifier	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
ExtraTreeClassifier	0.57	0.56	0.57	0.58	0.59	0.60	0.61	0.62	0.62	0.64	0.64	0.65	0.65	0.66	0.67	0.69	0.69	0.70	0.70	0.73	0.72	0.74	0.75	0.76	0.76	0.77	0.77	0.78	0.78	0.79	0.79	0.80	0.81	0.81	0.83	0.83	0.83	0.84	0.82	0.82
ExtraTreesClassifier	0.58	0.61	0.63	0.65	0.66	0.68	0.69	0.71	0.71	0.72	0.74	0.74	0.75	0.76	0.77	0.78	0.78	0.79	0.81	0.81	0.82	0.83	0.84	0.85	0.85	0.86	0.86	0.87	0.87	0.88	0.89	0.89	0.90	0.90	0.91	0.91	0.91	0.91	0.91	0.91
GaussianNB	0.59	0.60	0.60	0.61	0.61	0.61	0.61	0.62	0.61	0.61	0.60	0.60	0.60	0.61	0.62	0.62	0.63	0.63	0.64	0.64	0.65	0.65	0.66	0.66	0.67	0.67	0.68	0.68	0.69	0.69	0.69	0.69	0.69	0.70	0.70	0.70	0.70	0.70	0.70	0.70
KNeighborsClassifier	0.58	0.60	0.61	0.61	0.62	0.63	0.64	0.65	0.65	0.65	0.66	0.66	0.66	0.67	0.67	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.72	0.73	0.73	0.73	0.74	0.74	0.74	0.75	0.75	0.76	0.76	0.76	0.77	0.77	0.77	0.77	0.77	0.77
LGBMClassifier	0.64	0.65	0.67	0.67	0.69	0.70	0.72	0.73	0.73	0.74	0.75	0.76	0.76	0.77	0.78	0.79	0.80	0.81	0.81	0.82	0.83	0.84	0.85	0.85	0.86	0.87	0.87	0.87	0.88	0.89	0.90	0.90	0.91	0.91	0.91	0.92	0.92	0.92	0.92	0.92
LinearDiscriminantAnalysis	0.62	0.63	0.65	0.65	0.66	0.67	0.67	0.68	0.68	0.68	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.73	0.74	0.75	0.75	0.75	0.75	0.75	0.76	0.76	0.77	0.77	0.77	0.78	0.78	0.79	0.80	0.80	0.80	0.80	0.80
LinearSVC	0.63	0.64	0.65	0.65	0.66	0.67	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.74	0.74	0.74	0.76	0.76	0.77	0.78	0.79	0.79	0.80	0.80	0.81	0.82	0.82	0.83	0.84	0.84	0.84	0.84	0.86	0.86	0.86	0.86	0.86
LogisticRegression	0.63	0.64	0.66	0.66	0.67	0.67	0.68	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.74	0.74	0.75	0.75	0.77	0.77	0.78	0.79	0.80	0.81	0.82	0.82	0.82	0.83	0.84	0.85	0.85	0.85	0.85	0.86	0.87	0.87	0.87	0.87	0.87
NearestCentroid	0.62	0.63	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.66	0.66	0.66	0.67	0.67	0.67	0.68	0.68	0.68	0.69	0.69	0.70	0.70	0.70	0.70	0.70	0.71	0.71	0.71	0.71	0.72	0.72	0.72	0.72	0.72	0.72	0.72
NuSVC	0.59	0.60	0.62	0.63	0.64	0.65	0.66	0.68	0.69	0.70	0.71	0.72	0.72	0.73	0.75	0.77	0.78	0.78	0.79	0.80	0.80	0.80	0.81	0.82	0.82	0.83	0.83	0.83	0.83	0.84	0.84	0.84	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
PassiveAggressiveClassifier	0.55	0.55	0.59	0.60	0.60	0.60	0.58	0.61	0.62	0.63	0.63	0.63	0.64	0.65	0.67	0.62	0.62	0.62	0.62	0.62	0.71	0.75	0.76	0.68	0.69	0.68	0.73	0.68	0.79	0.80	0.78	0.81	0.79	0.73	0.73	0.74	0.74	0.74	0.75	0.75
Perceptron	0.52	0.53	0.55	0.61	0.58	0.61	0.59	0.64	0.60	0.62	0.63	0.64	0.61	0.65	0.68	0.65	0.70	0.70	0.71	0.72	0.74	0.75	0.75	0.71	0.77	0.78	0.78	0.75	0.78	0.80	0.78	0.78	0.73	0.80	0.80	0.82	0.80	0.82	0.80	0.80
QuadraticDiscriminantAnalysis	0.58	0.56	0.59	0.57	0.55	0.58	0.60	0.58	0.59	0.60	0.60	0.60	0.61	0.64	0.64	0.64	0.63	0.64	0.65	0.65	0.67	0.66	0.67	0.68	0.67	0.69	0.68	0.68	0.69	0.70	0.70	0.70	0.69	0.70	0.71	0.70	0.71	0.71	0.71	0.71
RandomForestClassifier	0.61	0.64	0.65	0.67	0.68	0.69	0.71	0.72	0.73	0.74	0.74	0.75	0.75	0.76	0.77	0.79	0.79	0.80	0.81	0.82	0.83	0.83	0.85	0.85	0.86	0.87	0.87	0.88	0.88	0.89	0.89	0.90	0.91	0.91	0.91	0.91	0.91	0.92	0.91	0.91
RidgeClassifier	0.62	0.63	0.65	0.65	0.66	0.67	0.67	0.68	0.68	0.68	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.72	0.72	0.73	0.73	0.74	0.75	0.75	0.75	0.75	0.75	0.76	0.76	0.77	0.77	0.77	0.78	0.78	0.79	0.80	0.80	0.80	0.80	0.80
RidgeClassifierCV	0.62	0.63	0.65	0.65	0.66	0.67	0.67	0.68	0.68	0.68	0.68	0.69	0.69	0.70	0.70	0.71	0.71	0.71	0.72	0.73	0.73	0.74	0.75	0.75	0.75	0.75	0.75	0.76	0.76	0.77	0.77	0.77	0.78	0.78	0.79	0.80	0.80	0.80	0.80	0.80
SGDClassifier	0.60	0.62	0.63	0.63	0.63	0.65	0.67	0.68	0.67	0.68	0.69	0.69	0.69	0.70	0.70	0.73	0.72	0.72	0.75	0.76	0.76	0.77	0.79	0.79	0.80	0.80	0.81	0.81	0.81	0.82	0.83	0.84	0.85	0.85	0.85	0.86	0.86	0.86	0.87	0.86
SVC	0.63	0.64	0.66	0.66	0.67	0.68	0.70	0.71	0.71	0.72	0.72	0.73	0.74	0.74	0.76	0.77	0.77	0.78	0.79	0.80	0.81	0.81	0.82	0.83	0.84	0.85	0.85	0.85	0.86	0.86	0.87	0.88	0.88	0.89	0.89	0.89	0.89	0.89	0.89	0.89
XGBClassifier	0.63	0.64	0.65	0.66	0.67	0.69	0.71	0.71	0.72	0.73	0.74	0.75	0.75	0.76	0.77	0.78	0.79	0.80	0.81	0.82	0.82	0.83	0.84	0.85	0.86	0.87	0.87	0.87	0.88	0.88	0.90	0.89	0.90	0.91	0.91	0.92	0.91	0.92	0.91	0.91

4.7 Summary

The MLAs with the highest scores for each of the different metrics (accuracy, balanced accuracy, F1 Score, and ROC AUC score) were identified for each of the 40 weeks. Based on this, the MLA with the best overall performance for each of the 40 weeks, was highlighted. From Tables 4.18 to 4.21, it was observed that there were 4 MLAs, namely LGBMClassifier, AdaBoostClassifier, RandomForestClassifier, and XGBClassifier, that had the highest scores across any of the four metrics, over the 40 weeks. RO2 and RO3 have been met in this chapter. The next chapter will provide a discussion of the results of the experiments conducted for this study.

Chapter 5: A Discussion on the Results of the Experiment

5.1 Introduction

This chapter discusses the findings of the experiments that used ensemble ML techniques for the early prediction SAR in a VLE. This study focused on training forty different ML predictive models, one for each week of the semester, using twenty-five different MLAs. Each model was trained using students' demographic data combined with data from their weekly interactions with a VLE. Based on the training results, four classifiers, namely AdaBoostClassifier, LGBMClassifier, RandomForestClassifier, and XGBClassifier were selected for hyperparameter optimization. A soft voting ensemble classifier was then used to combine the four classifiers in order to create an improved classifier with better prediction accuracy than any of the four individual classifiers.

5.2 Evaluation of the Machine Learning Models

To evaluate the ML models, different metrics were used, namely accuracy, balanced accuracy, F1 score, and ROC AUC score. A summary of the prediction scores is given in Table 4.18 (accuracy), Table 4.19 (balanced accuracy), Table 4.20 (F1 Score), and Table 4.21 (ROC AUC score), with the highest score for each week highlighted in green.

Based on the prediction scores in the four tables, the MLA with the highest prediction score for each week, for each of the four metrics, was extracted and summarised in Table 5.1. The MLA that had the highest prediction score across all four metrics for a particular week, was then chosen as the best performing MLA for that week.

Table 5.1: Summary of machine learning algorithms chosen over 40 weeks.

Week	Accuracy	Balanced Accuracy	F1 Score	AUC ROC Score	Chosen Algorithm
0	AdaBoostClassifier	AdaBoostClassifier	AdaBoostClassifier	AdaBoostClassifier	AdaBoostClassifier
1	AdaBoostClassifier	AdaBoostClassifier	AdaBoostClassifier	AdaBoostClassifier	AdaBoostClassifier
2	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
3	AdaBoostClassifier	AdaBoostClassifier	AdaBoostClassifier	AdaBoostClassifier	AdaBoostClassifier
4	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
5	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
6	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
7	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
8	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
9	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
10	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
11	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
12	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
13	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
14	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
15	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
16	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
17	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
18	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
19	RandomForestClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
20	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
21	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
22	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
23	RandomForestClassifier	LGBMClassifier	RandomForestClassifier	LGBMClassifier	LGBMClassifier
24	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
25	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
26	RandomForestClassifier	XGBClassifier	XGBClassifier	XGBClassifier	XGBClassifier
27	RandomForestClassifier	RandomForestClassifier	RandomForestClassifier	RandomForestClassifier	RandomForestClassifier
28	RandomForestClassifier	RandomForestClassifier	RandomForestClassifier	RandomForestClassifier	RandomForestClassifier
29	RandomForestClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
30	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
31	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
32	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
33	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
34	RandomForestClassifier	XGBClassifier	XGBClassifier	XGBClassifier	XGBClassifier
35	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
36	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
37	RandomForestClassifier	LGBMClassifier	RandomForestClassifier	LGBMClassifier	LGBMClassifier
38	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier
39	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier	LGBMClassifier

Using the results of the chosen algorithm column from Table 5.1, the prediction scores of AdaBoostClassifier, LGBMClassifier, RandomForestClassifier, and XGBClassifier algorithms, were graphed for each of the four different metrics. These graphs are depicted in Figure 5.1, Figure 5.2, Figure 5.3, and Figure 5.4.

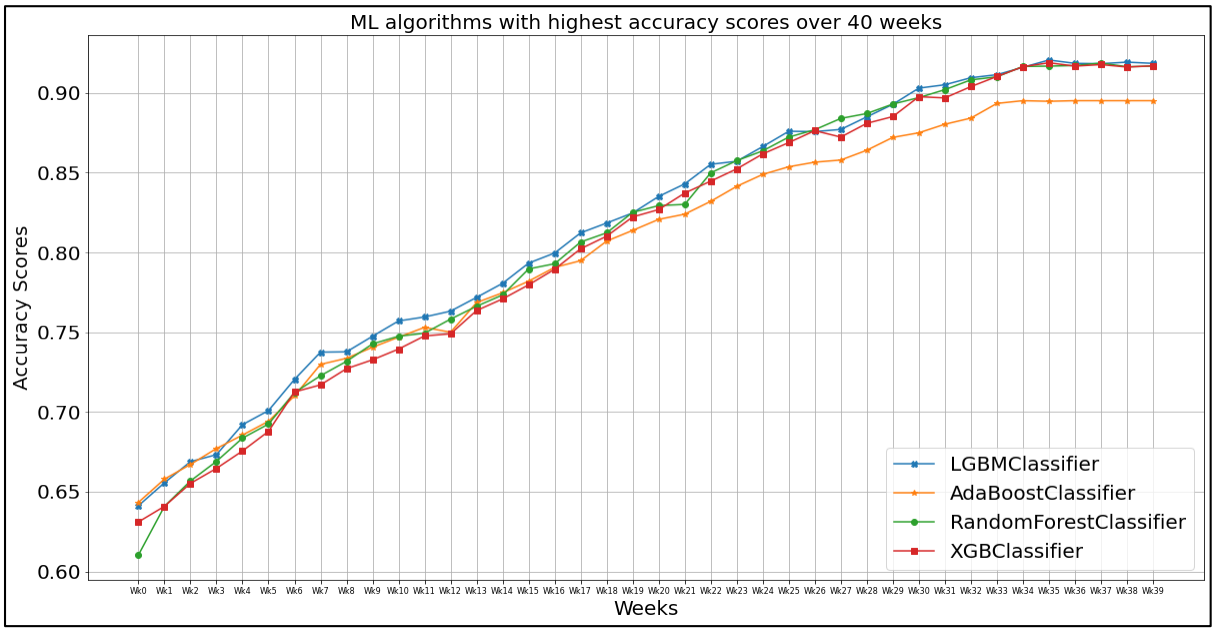

Figure 5.1: Accuracy scores of best performing machine learning algorithms.

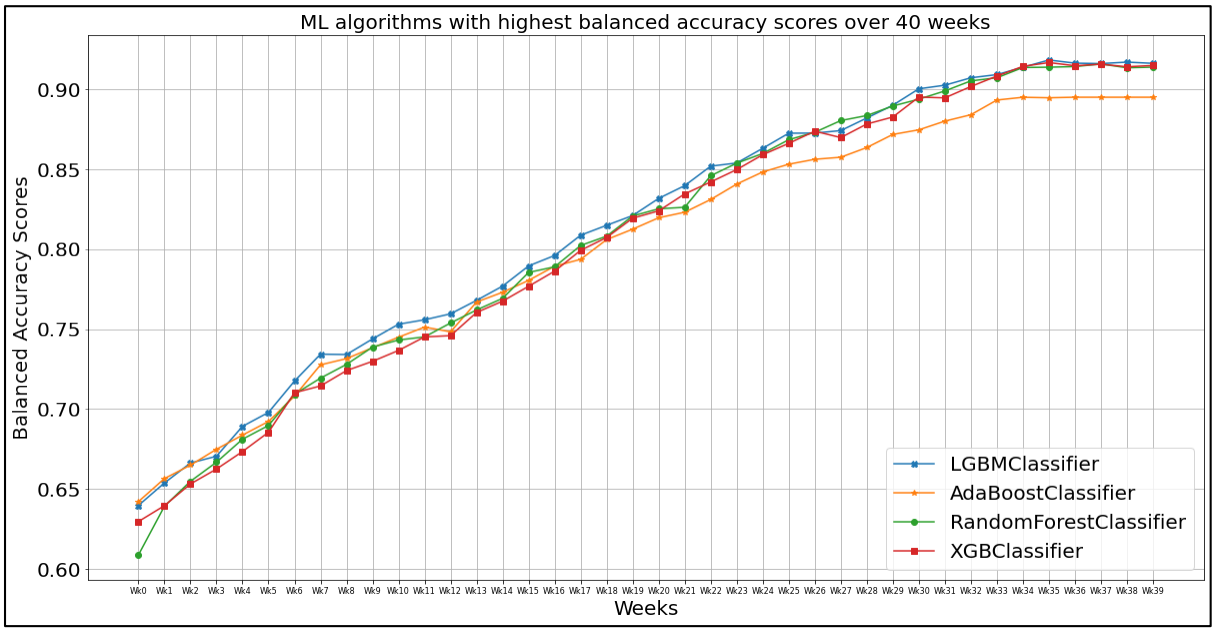

Figure 5.2: Balanced accuracy scores of best performing machine learning algorithms.

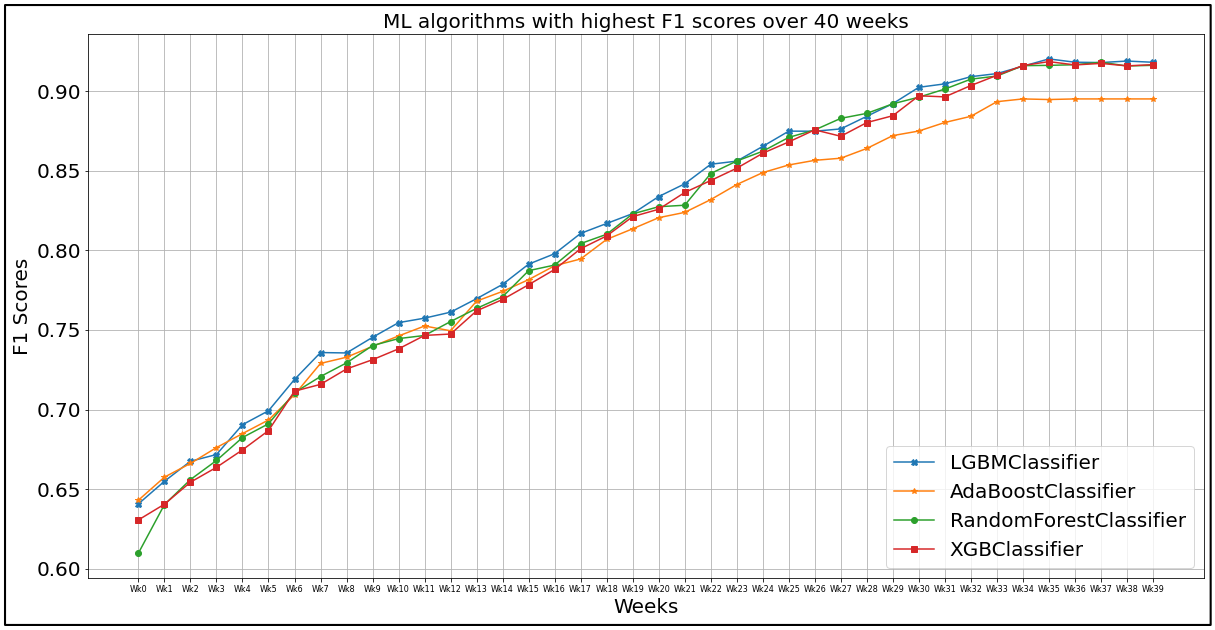

Figure 5.3: F1 Scores of best performing algorithms.

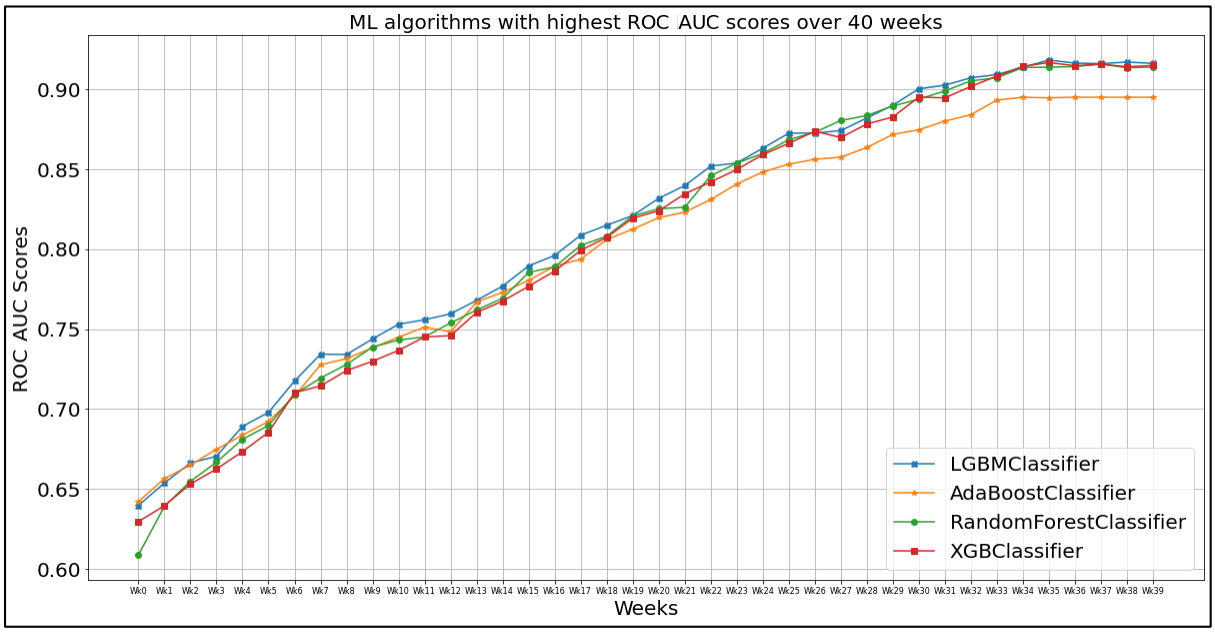

Figure 5.4: ROC AUC scores of best performing algorithms.

5.3 A Discussion of the Findings of the Experiment

Based on the results of the prediction using 4 MLAs, the following observations can be made:

- Prediction scores using each of the 4 metrics (accuracy, balanced accuracy, F1 score, and ROC AUC score) ranged from 64% to 92%.

- The prediction score of 64% was observed in week 0, before the course started, but in a week where students had access to VLE resources. Thus, with just relatively minimal engagement time with the VLE resources, a much better prediction score than a baseline no-skill classifier, was achieved.
- Four weeks after the course started, the ML model was able to predict SAR with 70% accuracy. After 8 weeks, the accuracy improved to 80% and progressively improved each week.
- Prediction scores progressively improved from 64% in week 0 to 92% in week 39. The only change in the data supplied to each predictive model on a weekly basis, was the addition of the weekly VLE interactions from all previous weeks up to and including the current week. One can infer from this that the addition of more weekly VLE interaction data improved the prediction scores.
- Since the dataset was fairly balanced, the prediction scores of the four metrics followed a fairly similar pattern over the 40 weeks.
- For the 40 weeks, the LGBMClassifier MLA was the best performing for 33 weeks, followed by the AdaBoostClassifier MLA for 3 weeks, and finally the RandomForestClassifier and XGBClassifier MLAs for 2 weeks each.

The MLAs with the highest scores for each of the different metrics (accuracy, balanced accuracy, F1 Score, and ROC AUC score) were identified for each of the 40 weeks. Based on this, the MLA that can be used for prediction for each of the 40 weeks, was chosen. While the LGBMClassifier was best performing for 33 out of the 40 weeks, given that the performance of the four MLAs differed by less than 2%, any of the four algorithms can be used to effectively predict SAR.

While individual classifiers were shown to be effective in the early prediction of SAR, hyperparameter tuning of the individual base classifiers, together with the creation of a voting classifier ensemble method improved predictive performance even further. This is discussed in the next section.

5.4 Improved Prediction Using Voting Classifier Ensemble Method

Although the findings of the experiment as discussed in the previous section show that any of the four classifiers can effectively be used for the early prediction of SAR, this study was extended through the use of ensemble learning to create an improved ML-based model. To achieve this, hyperparameter optimization was done using Random Search to tune appropriate

hyperparameters of the four base classifiers, namely LGBMClassifier, AdaBoostClassifier, RandomForestClassifier and XGBClassifier. A soft voting ensemble classifier was then used to combine the four classifiers in order to create an improved classifier with better prediction accuracy than any of the four individual classifiers. This will be discussed next.

5.4.1 Hyperparameter Tuning of the Base Classifiers

In this section, hyperparameter tuning of the four base classifiers is discussed.

5.4.1.1 RandomForestClassifier

The Python code snippet 11 shows the tuning of the *n_estimators, max_depth* and *max_features* hyperparameters of the RandomForestClassifier using 10-fold cross validation.

```
param_space = {'max_depth': [5,7,10,15], 'max_features': ['auto', 'sqrt'],'n_estimators': [100,500,1000]}
rf_clf = RandomizedSearchCV(estimator = RandomForestClassifier(),
                            param_distributions=param_space, n_iter=40,scoring='accuracy',
                            n_jobs=-1, cv=10,refit=True, return_train_score=True)
rf_clf.fit(X_train, y_train)
best_params_rf = rf_clf.best_params_
```

Code Snippet 11: Hyperparameter tuning for RandomForestClassifier.

The best hyperparameter values were:

{'*n_estimators*': 1000, '*max_features*': 'auto', '*max_depth*': 5}

5.4.1.2 XGBClassifier

The Python code snippet 12 shows the tuning of the *n_estimators, max_depth* and *learning_rate* hyperparameters of the XGBClassifier using 10-fold cross validation.

```
param_space = {'max_depth': [5,7,10,15], 'learning_rate': [0.001, 0.1],'n_estimators': [100,500,1000]}

xgb_clf = RandomizedSearchCV(estimator = XGBClassifier(),
                             param_distributions=param_space, n_iter=40,scoring='accuracy',
                             n_jobs=-1, cv=10,refit=True, return_train_score=True)
xgb_clf.fit(X_train, y_train)
best_params_xgb = xgb_clf.best_params_
```

Code Snippet 12: Hyperparameter tuning for XGBClassifier.

The best hyperparameter values were:

{'*n_estimators*': 500, *'max_depth*': 7, '*learning_rate'*: 0.001}

5.4.1.3 LGBMClassifier

The Python code snippet 13 shows the tuning of the *num_leaves, n_estimators* , *max_depth, bagging_frequency, and bagging_fraction* hyperparameters of the LGBMClassifier using 10-fold cross validation.

```
param_space = {'max_depth': [5,7,10,15],'bagging_fraction': [0.5, 0.8],'bagging_frequency': [5, 8],
               'num_leaves': (1200, 1550),'n_estimators': [100,500,1000]}
lgbm_clf = RandomizedSearchCV(estimator = LGBMClassifier(),
                              param_distributions=param_space, n_iter=60,scoring='accuracy',
                              n_jobs=-1, cv=10,refit=True, return_train_score=True)
lgbm_clf.fit(X_train, y_train)
best_params_lgbm = lgbm_clf.best_params_
```

Code Snippet 13: Hyperparameter tuning for LGBMClassifier.

The best hyperparameter values were:

{'*num_leaves*': 1200, '*n_estimators*': 1000, '*max_depth*': 10, '*bagging_frequency*': 8, '*bagging_fraction*': 0.5}

5.4.1.4 AdaBoostClassifier

The Python code snippet 14 shows the tuning of the *n_estimators* and *learning_rate* hyperparameters of the AdaBoostClassifier using 10-fold cross validation.

```
param_space = {'learning_rate': [0.001, 0.1],'n_estimators': [100,500,1000]}

ab_clf = RandomizedSearchCV(estimator = AdaBoostClassifier(),
                            param_distributions=param_space, n_iter=40,scoring='accuracy',
                            n_jobs=-1, cv=10,refit=True, return_train_score=True)
ab_clf.fit(X_train, y_train)
best_params_ab = ab_clf.best_params_
```

Code Snippet 14: Hyperparameter tuning for AdaBoostClassifier.

The best hyperparameter values were:

{'*n_estimators*': 1000, '*learning_rate*': 0.1}

5.4.2 Evaluation of the Voting Classifier Ensemble method

In this section, the voting classifier ensemble method prediction results, created from the hyperparameter-tuned classifiers, is discussed. The 4 classifiers' and the voting classifier's accuracy scores for 40 weeks, together with the graph of these scores, show that the voting classifier performs better overall than the individual classifiers. Code snippet 15 shows the voting ensemble method used to predict SAR over 40 weeks using 40 different datasets.

```
AccList =[]
for wk in range(40):
    fname= 'W'+str(wk)+'.csv'
    dfScaled = pd.read_csv(fname,header=0, index_col=0)
    X = dfScaled.drop(['risk_status'],axis=1)
    y = dfScaled['risk_status']
    clf1 = RandomForestClassifier(n_estimators=100,random_state=1, max_depth = 5, max_features = 'auto')
    clf2 = XGBClassifier(n_estimators=500,random_state=1,learning_rate=0.001,max_depth=7)
    clf3 = LGBMClassifier(n_estimators=1000,random_state=1,max_depth=10, num_leaves=1200,
                          bagging_frequency=8,bagging_fraction=0.5)
    clf4 = AdaBoostClassifier(n_estimators=100,random_state=1,learning_rate = 0.1)
    clf5 = VotingClassifier(estimators=[('RF', clf1),('XGB', clf2),('LGBM', clf3),('AB', clf4)], voting='soft')
    for clf in (clf1,clf2,clf3,clf4,clf5):
        acc = cross_val_score(clf, X, y, scoring ='accuracy',cv=10).mean()
        AccList.append(acc)
```

Code Snippet 15: Using the voting classifier ensemble method to predict SAR.

Each of the 40 datasets which was extensively pre-processed, comprises the time-invariant biographical data and the cumulative weekly VLE interactions data for each of the 40 weeks. Each of the 4 base classifiers contained hyperparameters that were tuned, as discussed in section 5.4.1. These classifiers were used to create the voting classifier ensemble with 10-fold

cross validation being used to evaluate the predictive models. The classifiers' accuracy scores for each of the 40 weeks are shown in Table 5.2.

Table 5.2: Accuracy scores for 4 hyperparameter-turned classifiers and voting classifier ensemble method over 40 weeks.

	ML Algorithm	Wk0	Wk1	Wk2	Wk3	Wk4	Wk5	Wk6	Wk7	Wk8	Wk9	Wk10	Wk11	Wk12	Wk13	Wk14	Wk15	Wk16	Wk17	Wk18	Wk19	Wk20	Wk21	Wk22	Wk23	Wk24	Wk25	Wk26	Wk27	Wk28	Wk29	Wk30	Wk31	Wk32	Wk33	Wk34	Wk35	Wk36	Wk37	Wk38	Wk39
0	AdaBoostClassifier	0.66	0.67	0.69	0.70	0.71	0.72	0.73	0.74	0.75	0.75	0.76	0.77	0.77	0.78	0.78	0.79	0.80	0.81	0.81	0.82	0.83	0.84	0.84	0.85	0.86	0.86	0.87	0.87	0.87	0.88	0.88	0.88	0.89	0.90	0.90	0.90	0.90	0.90	0.90	0.90
1	LGBMClassifier	0.65	0.66	0.67	0.68	0.70	0.71	0.73	0.74	0.74	0.75	0.76	0.76	0.77	0.78	0.79	0.80	0.80	0.82	0.82	0.83	0.84	0.85	0.86	0.86	0.87	0.88	0.88	0.88	0.89	0.90	0.91	0.91	0.91	0.92	0.92	0.93	0.92	0.92	0.92	0.92
2	RandomForestClassifier	0.66	0.68	0.70	0.71	0.72	0.73	0.73	0.74	0.75	0.76	0.77	0.78	0.78	0.79	0.80	0.81	0.81	0.82	0.83	0.84	0.85	0.85	0.87	0.87	0.88	0.89	0.89	0.89	0.90	0.90	0.91	0.91	0.91	0.91	0.92	0.92	0.92	0.92	0.92	0.92
3	XGBClassifier	0.66	0.67	0.70	0.70	0.71	0.73	0.73	0.74	0.75	0.76	0.77	0.78	0.78	0.79	0.80	0.81	0.81	0.82	0.83	0.84	0.85	0.86	0.87	0.88	0.88	0.89	0.89	0.89	0.90	0.90	0.90	0.91	0.91	0.91	0.92	0.92	0.92	0.92	0.92	0.92
4	VotingEnsemble	0.66	0.68	0.69	0.70	0.71	0.73	0.74	0.74	0.75	0.77	0.78	0.78	0.79	0.80	0.80	0.81	0.82	0.83	0.84	0.85	0.86	0.87	0.88	0.88	0.89	0.89	0.89	0.89	0.90	0.91	0.92	0.92	0.92	0.93	0.93	0.93	0.93	0.93	0.93	0.93

Figure 5.5 depicts the accuracy scores of the AdaBoostClassifier, LGBMClassifier, RandomForestClassifier, and XGBClassifier classifiers together with the voting classifier ensemble method.

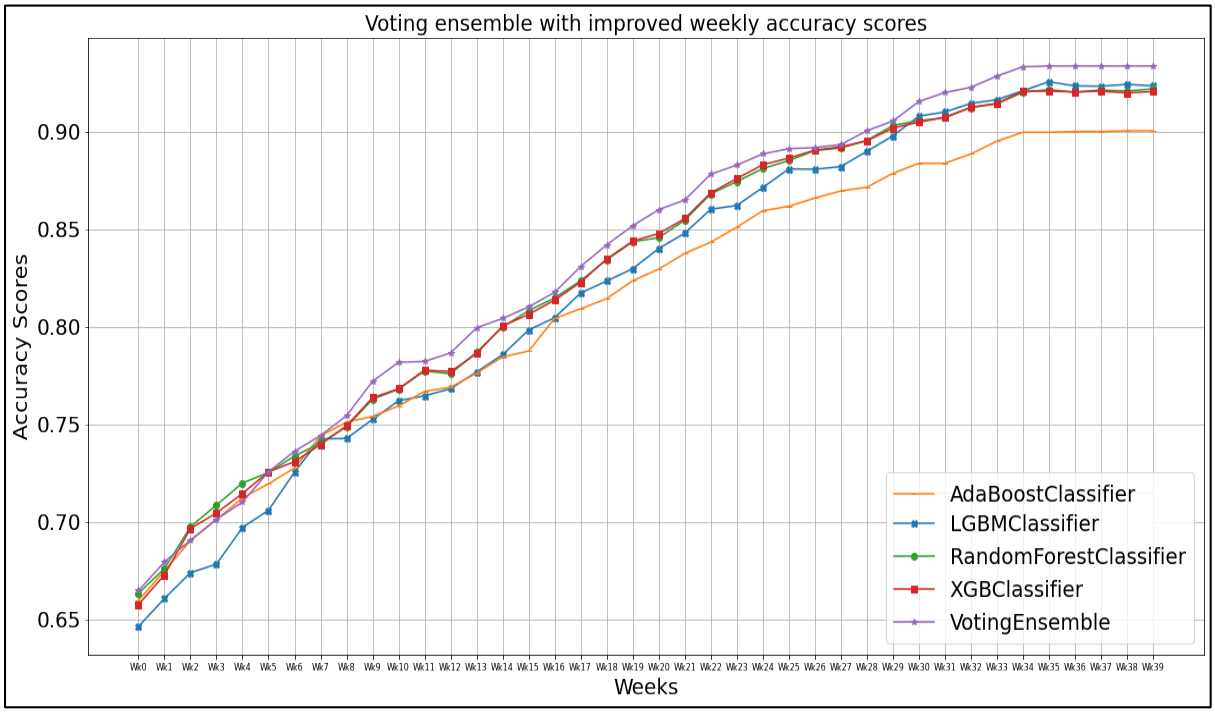

Figure 5.5: Accuracy scores of 4 classifiers together with voting classifier ensemble method.

The results show that after hyperparameter tuning of the AdaBoostClassifier, LGBMClassifier, RandomForestClassifier, and XGBClassifier classifiers, accuracy scores improved, as can be observed from Table 5.2.

Furthermore, the voting classifier ensemble method outperformed the individual classifiers overall over 40 weeks as can be observed from the accuracy scores in Table 5.2 and the graph in Figure 5.5.

5.5 Summary

This study set out to determine if ML-based predictive modelling techniques can be used for the early identification of SAR. The results and findings discussed in this chapter show that a ML-based predictive modelling approach, using the MLAs identified in the experiment, can indeed be used for the early identification of SAR. In this study, this was done on a weekly basis. RO3 has therefore been met in this chapter. Hyperparameter tuning of the individual

base classifiers, together with the creation of a voting classifier ensemble method improved predictive performance. The next chapter concludes the dissertation with a summary and further recommendations.

Chapter 6: Conclusion, Limitations and Future Research

6.1 Introduction

This chapter will revisit the research objectives stated in Chapter 1 and show how they were achieved during the course of this research. The limitations will be discussed, together with recommendations for further research, and the contribution towards early identification of SAR, and to the field of ML, will be highlighted.

In this study, a major point of departure from the vast majority of studies found in literature, was to exclude student test and examination results in the ML modelling process. While this may seem counter-intuitive, since a student who fails may be a strong candidate for being at risk, the reason for excluding test and examination results is as follows: tests are often written at least a few weeks into the course. In some instances, tests may be delayed even further. Thus, relying on test marks to primarily identify SAR would mean that SAR very often are identified several weeks after the course has started. This would delay any early interventions to assist the affected students. The findings of the study have shown that early identification of SAR can be done using attributes other than test marks.

Another point of departure was to create predictive ML models for weekly predictions of SAR. The usual way of attempting to identify SAR using the manual method, is to use assessment results. When a student fails a test, they may then be identified as being at risk. This meant that identifying SAR was neither early nor frequent enough. The literature shows that even though various authors used ML techniques to identify SAR, one of the shortcomings of the techniques used was that training was usually done on data from an entire semester, as opposed to creating and training multiple MLAs on data that is available on a weekly basis. This study instead focused on conducting predictive ML modelling for each of the forty weeks that made up a semester. The data used to train the ML models was the historical data available for a specific week and not just the same set of data for the entire semester. This would enable more accurate predictions of SAR.

The predictive ML modelling techniques presented in Chapter 4 and discussed in Chapter 5 could provide baseline information about SAR on a weekly basis. This in turn could allow for earlier interventions, that would otherwise not be possible through traditional statistical methods, or even using a single ML model that uses one full set of data for the entire semester of forty weeks.

6.2 Answering the Research Question

This study was guided by the following research question.

Research Question: What ML ensemble can be used for early and accurate prediction of SAR using students' demographic and weekly online Virtual Learning Environment (VLE) data?

To answer the research question, the following research objectives (ROs) were formulated.

Research Objectives:

RO1: To conduct a comparative performance analysis of MLAs used in predicting SAR.

RO2: To develop a MLA-based SAR prediction model using an ensemble of best-performing MLAs.

RO3: To experimentally validate this model using demographic and weekly online VLE data.

6.2.1 Research Objective 1: To conduct a Comparative Performance Analysis of MLAs used in Predicting SAR

A review of the literature in chapter 2 shows various ML techniques being used in different studies conducted to identify SAR at different HEIs, in different countries. Since the data varied from study to study, a direct comparison between the different studies could not be made. Furthermore, from the literature reviewed, the studies that used ML to identify SAR varied with regards to the MLAs that were classified as 'best' in terms of accuracy. Simply put, based on the literature, no one specific ML algorithm performed best in all of the different studies. This may be attributed to the *No Free Lunch* theorem from Wolpert and Macready (1997), which argues that there is no single classification algorithm that can outperform every other classification algorithm, in every problem area.

The literature provided some guidance on the ML models and techniques that may be suitable for this study. While the datasets and features may have varied across the different studies identified in the literature, they did provide some direction that could be used as a baseline for this study. However, since the performance of the different MLAs varied across the studies identified in the literature, this researcher chose to include twenty-five different MLAs for

training, to ensure diversity and avoid, or minimize, bias towards specific algorithms. The best performing MLAs, based on the results of four different metrics (accuracy, balanced accuracy, F1 score, and ROC AUC) for each of the forty weeks, was then chosen as baseline classifiers towards developing a MLA-based SAR prediction model using an ensemble of these best-performing MLAs.

While reviewing the literature, it was found that studies either considered demographic data in addition to study data, or pointed to a need to consider demographic data in order to identify or understand the reasons for SAR. This study used both demographic data and VLE data in order to identify SAR.

6.2.2 Research Objective 2: To Develop a MLA-based SAR Prediction Model Using an Ensemble of Best-Performing MLAs

Following the ML methodology, as outlined in section 3.4 of Chapter 3, experiments were conducted, as discussed in Chapter 4, to identify SAR on a weekly basis, for forty weeks of a semester. The performance of ML models depends greatly on the quality and format of the raw student data.

The raw data requires pre-processing so it can be transformed into a form that can be used for ML. In real-life situations, the data can be quite messy and may thus require a lot of pre-processing, before it can be used in ML. A rigorous search was conducted for an appropriate dataset, which was located and extensively pre-processed for ML. The pre-processing stage entailed, but was not limited to, performing an exploratory analysis of the data, deleting or updating rows (or columns) that contained missing data, converting all non-numeric data to numeric data, scaling the data, and engineering new features or attributes from existing student attributes. ML metrics, such as accuracy, balanced accuracy, F1 score, and ROC AUC score, were discussed in the context of their use in evaluating the performance of the MLAs. The pre-processed and transformed data was used as input to the ML models for identifying SAR.

The ML modelling process of this study entailed modelling and prediction of SAR on a weekly basis for forty weeks of a semester. This entailed training forty different ML predictive models, one for each week of the semester, using twenty-five different MLAs. Each model was trained using students' demographic data, combined with data from their weekly interactions with a VLE. For each week, starting at week 0, the model was supplied with students' demographic data (which remained the same for each of the forty weeks), together with data from the

students' VLE interaction from all of the preceding weeks, up to and including the current week. For instance, for week 2, VLE training data was supplied to the model as data from week 0, week 1, and week 2. This process was repeated for each of the forty weeks. The predictive capability of each model was then evaluated using the four different performance metrics, namely accuracy, F1 score, and ROC AUC score.

The MLAs with the highest scores for each of the different metrics (accuracy, balanced accuracy, F1 Score, and ROC AUC score), were identified for each of the forty weeks. The best-performing MLAs were then selected as base classifiers. Hyperparameter optimization was done using Random Search to tune appropriate hyperparameters of the four base classifiers, namely LGBMClassifier, AdaBoostClassifier, RandomForestClassifier and XGBClassifier. A soft voting ensemble classifier was then used to combine the four classifiers in order to create an improved classifier with better prediction accuracy than any of the four individual classifiers.

6.2.3 Research Objective 3: To Experimentally Validate this Model Using Demographic and Weekly Online VLE Data

This research objective set out to evaluate the performance of twenty-five MLAs that were applied to the dataset called OULAD, over a forty-week period, to identify base classifiers from the best-performing classifiers. There were thus forty ML experiments, one for each week. Based on the results of the experiments outlined in chapter 4, there were four out of the twenty-five candidate MLAs that had the highest scores across any of the four metrics, over the forty weeks.

As shown in Table 5.1 of chapter 5, and reflected in the graphs in Figure 5.1 to Figure 5.4, LGBMClassifier was the best performing algorithm in 33 of the 40 weeks. What is interesting to note, is that even in the weeks in which one of the other algorithms performed better, the LGBMClassifier performed almost as well, with a difference of 1% or less, compared to the top performing algorithm for that week. A further observation from Table 4.18 to Table 4.21 of chapter 4, is that in the weeks in which the LGBMClassifier performed the best, the other three algorithms performed almost as well, with a difference of 2% or less from the top performing algorithm for that week. From these observations, one can argue that any one of these four MLAs would be a worthy candidate to be used to identify SAR. However, these MLAs were then selected as base classifiers of a soft voting ensemble classifier for improved

predictions. Hyperparameter optimization was done using Random Search to tune appropriate hyperparameters of the four base classifiers using 10-fold cross validation. A soft voting ensemble classifier was then used to combine the four classifiers in order to create an improved classifier with better prediction accuracy than any of the four individual classifiers.

In meeting the three research objectives set, the research question ***"What ML ensemble can be used for early and accurate prediction of SAR using students' demographic and weekly online Virtual Learning Environment (VLE) data?"*** has been answered as follows:
The LGBMClassifier, AdaBoostClassifier, RandomForestClassifier, and XGBClassifier were determined to be the most effective classifiers to be used as base classifiers for a soft voting ensemble classifier for early and accurate prediction of SAR using students' demographic and weekly online Virtual Learning Environment (VLE) data. This further strengthens the main aim of the study, which was to employ ML-based predictive modelling techniques in early and accurate prediction of SAR at a HEI using students' demographic and weekly online VLE data.

6.3 Contribution of the Study

Given the perennial issue of SAR and its consequent negative impact on HEIs, any interventions that can help to minimize the SAR problem would be welcome by the HEIs.

The intended contribution of this study towards employing ML-based predictive modelling techniques in early and accurate prediction of SAR at a HEI are enunciated as follows:

a) Contribution to theory: This contribution is derived from answering the research question "What ML ensemble can be used for early and accurate prediction of SAR using students' demographic and weekly online Virtual Learning Environment (VLE) data?". Through the construction of a validated MLA-based SAR predictive model, this study found that a soft voting ensemble of four MLAs, namely AdaBoostClassifier, LGBMClassifier, RandomForestClassifier, and XGBClassifier, can be used for early and accurate prediction of SAR using students' demographic and weekly online Virtual VLE data.
b) Contribution to methodology: The validated methodology of this study can be reused by other researchers.
c) Contribution to practice: The results of this study can be used in practice for the early prediction of SAR.

6.4 Limitations

The dataset used for this study was from an online university in the UK. Due to ethics and data privacy issues, the dataset called OULAD was used. Since the dataset was designed for research purposes, it was completely anonymized to avoid any ethical issues. A search of the literature found no local dataset suitable for this study. In order to extend this research for a HEI in South Africa, local data will have to be collected and used.

A second limitation of this study is that the data for the online interactions had to be aggregated. The initial number of student interactions (or clicks) with the VLE resources was 106 552 780. This would have been computationally expensive to model within a reasonable timeframe. The individual daily interactions for each student had to then be reduced from 10 555 280 to 26 070, through the use of data aggregation techniques as discussed in section 4.2.2.2. The Python code snippets 3 and 5 which appear in appendix B, were used to generate the aggregated data reflected in Tables 4.11 and 4.12. While the data aggregation did not have an impact on the overall aim of being able to identify SAR using ML techniques, identifying which of the individual, different VLE interactions that may have had the greatest impact, was no longer possible.

A third limitation is the use of only the Random Search for the hyperparameter optimization. Due to hardware and time constraints, it was not feasible to use other hyperparameter search methods such as Grid Search which may have resulted in greater tuning of the hyperparameters.

6.5 Validity, Reliability and Generalizability of the Study

Internal validity of the study was achieved through the use of the ML pipeline and the results that were obtained, by using four different metrics to cross-reference the results. Reliability was achieved, by comparing the results for the four best algorithms over 40 experiments, and observing the closeness of the prediction results. External validity and generalizability were achieved in the use of unseen data for predictions during the ML process. If similar data is collected from a different source, the prediction scores are expected to be very similar to the current predictions.

6.6 Future Research

The results of this study and the knowledge gained provide the foundations for future research to build upon, by packaging the ML models and developing an application with a graphical user interface for ease of use. This model can then be deployed for use in identifying SAR in various HEI settings.

This study focused on using ML techniques for the early identification of SAR using students' demographic and VLE data. In order to compare the predictive performance of the different ML-based models over forty weeks, based on the demographic and VLE data, assessment marks were excluded as part of the training data. The reason for this exclusion was explained in section 6.1. However, for future research, assessment marks can be included in the training dataset of those weekly ML-based models, when the assessment marks become available.

The VLE data contained twenty different activity types as shown in Fig 4.2. However due to the reasons mentioned in section 6.4, the VLE data had to be aggregated. While the aggregation of the data was necessary for practical purposes, a granular analysis of the impact of individual VLE activities was no longer possible. Cloud-based graphics processing units (GPUs) and other powerful hardware, if available may eliminate the need for aggregating the VLE data.

Deep learning can be used to eliminate some of the data pre-processing by automating feature selection. Deep learning is in essence an ANN with at least three layers (Shrestha & Mahmood, 2019).

6.7 Conclusion

This study set out to determine if ML modelling techniques could be used for the early identification of SAR. The findings of the study show that a MLA-based SAR prediction model using an ensemble of best-performing MLAs ML has promising prospects of being able to be used for the early identification of SAR. The findings further show that identifying SAR can be done using demographic, as well as VLE interactions data, as opposed to just using assessment results. The latter is often the case, especially when using the manual or traditional statistical method of identifying SAR.

A review of the literature in Chapter 2 highlighted the impact of SAR on HEIs, the economy and students. Some of the key challenges of using traditional statistical approaches and current ML-based approaches to identify SAR were also identified. Based on these challenges, a MLA-based SAR prediction model using an ensemble of best-performing MLAs was proposed as an

alternative for the improved identification of SAR. This model was experimentally validated using demographic and weekly online VLE data. The results of this study can be used in practice for the early prediction of SAR.

References

Acharya, A., & Sinha, D. (2014). Early prediction of students performance using machine learning techniques. *International Journal of Computer Applications, 107*(1).

Adnan, M., Habib, A., Ashraf, J., Mussadiq, S., Raza, A., Abid, M., . . . Khan, S. (2021). Predicting at-Risk Students at Different Percentages of Course Length for Early Intervention Using Machine Learning Models. *IEEE Access, 9*, 7519-7539. doi:10.1109/ACCESS.2021.3049446

Africheck. (2016). Factsheet: Funding and the changing face of South Africa's public universities. Retrieved from https://africacheck.org/fact-checks/factsheets/factsheet-funding-and-changing-face-south-africas-public-universities

Agrawal, H., & Mavani, H. (2015). Student Performance Prediction using Machine Learning. *International Journal of Engineering Research and Technology*.

Ajibade, S.-S. M., Ahmad, N. B. B., & Shamsuddin, S. M. (2019). *Educational data mining: enhancement of student performance model using ensemble methods.* Paper presented at the IOP Conference Series: Materials Science and Engineering.

Ajjawi, R., Dracup, M., Zacharias, N., Bennett, S., & Boud, D. (2020). Persisting students' explanations of and emotional responses to academic failure. *Higher Education Research & Development, 39*(2), 185-199. doi:10.1080/07294360.2019.1664999

Al-Azawei, A., & Al-Masoudy, M. (2020). Predicting Learners' Performance in Virtual Learning Environment (VLE) based on Demographic, Behavioral and Engagement Antecedents. *International Journal of Emerging Technologies in Learning, 15*(9), 60-75.

Al-Obeidat, F., Tubaishat, A., Dillon, A., & Shah, B. (2017). Analyzing students' performance using multi-criteria classification. *Cluster Computing*, 1-10.

Ali, H., Salleh, M. N. M., Saedudin, R., Hussain, K., & Mushtaq, M. F. (2019). Imbalance class problems in data mining: A review. *Indonesian Journal of Electrical Engineering Computer Science, 14*(3), 1560-1571.

Aljohani, N., Fayoumi, A., & Hassan, S.-U. (2019). Predicting At-Risk Students Using Clickstream Data in the Virtual Learning Environment. *Sustainability, 11*, 7238. doi:10.3390/su11247238

Andrade, C. (2018). Internal, external, and ecological validity in research design, conduct, and evaluation. *Indian journal of psychological medicine, 40*(5), 498-499.

Asif, R., Merceron, A., & Pathan, M. K. (2014). Predicting student academic performance at degree level: a case study. *International Journal of Intelligent Systems and Applications, 7*(1), 49.

Azizah, E. N., Pujianto, U., Nugraha, E., & Darusalam. (2018, 26-28 Oct. 2018). *Comparative performance between C4.5 and Naive Bayes classifiers in predicting student academic performance in a Virtual Learning Environment.* Paper presented at the 2018 4th International Conference on Education and Technology (ICET).

Bali, R., Sarkar, D., & Sharma, T. (2018). *Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems*. New York: Apress.

Barrionuevo, G. O., Ríos, S., Williams, S. W., & Ramos-Grez, J. A. (2021). *Comparative Evaluation of Machine Learning Regressors for the Layer Geometry Prediction in Wire arc Additive manufacturing.* Paper presented at the 2021 IEEE 12th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT), Cape Town, South Africa.

Bayer, J., Bydzovská, H., Géryk, J., Obsivac, T., & Popelinsky, L. (2012). Predicting Drop-Out from Social Behaviour of Students. *International Educational Data Mining Society*.

Berens, J., Schneider, K., Görtz, S., Oster, S., & Burghoff, J. (2018). Early detection of students at risk–predicting student dropouts using administrative student data and machine learning methods.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research, 13*(2), 281-305.

Beyeler, M. (2017). *Machine Learning for OpenCV* (M. Sinha Ed. First ed.). Birhingham, UK: Packt Publishing Ltd.

Bhandari, P. (2021, 2 March 2021). An Introduction to Inferential Statistics. Retrieved from https://www.scribbr.com/statistics/inferential-statistics/

Bowie, P. (2019). *The English Indices of Deprivation 2019 (IoD2019)*. Retrieved from https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/835115/IoD2019_Statistical_Release.pdf

Bramer, M. (2013). *Principles of Data Mining*: Springer Publishing Company, Incorporated.

Bryman, A. (2012). *Social Research Methods* (4th ed.). Oxford; New York: Oxford University Press.

Butcher, Z. E. (2021). *Contract Information Extraction Using Machine Learning*. Retrieved from

Bzdok, D., Altman, N., & Krzywinski, M. (2018). Statistics versus machine learning. *Nature methods, 15*(4), 233.

Chai, K. E., & Gibson, D. (2015). Predicting the Risk of Attrition for Undergraduate Students with Time Based Modelling. *International Association for Development of the Information Society*.

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

Chen, J.-F., Hsieh, H.-N., & Do, Q. H. (2014). Predicting student academic performance: a comparison of two meta-heuristic algorithms inspired by cuckoo birds for training neural networks. *Algorithms, 7*(4), 538-553.

Chicco, D. (2017). Ten quick tips for machine learning in computational biology. *BioData Mining, 10*(1), 35. doi:10.1186/s13040-017-0155-3

Chui, K. T., Fung, D. C. L., Lytras, M. D., & Lam, T. M. (2020). Predicting at-risk university students in a virtual learning environment via a machine learning algorithm. *Computers in Human Behavior, 107*(2020), 105584. doi:https://doi.org/10.1016/j.chb.2018.06.032

Cichos, F., Gustavsson, K., Mehlig, B., & Volpe, G. (2020). Machine learning for active matter. *Nature Machine Intelligence, 2*(2), 94-103.

Coates, H. (2005). The value of student engagement for higher education quality assurance. *Quality in Higher Education, 11*(1), 25-36. doi:10.1080/13538320500074915

Cook, B., & Pullaro, N. (2010). College graduation rates: Behind the numbers.

Creswell, J. W., & Creswell, J. D. (2018). *Research design : qualitative, quantitative & mixed methods approaches* (5th ed. ed.). Los Angeles: SAGE.

Darst, B., Malecki, K., & Engelman, C. (2018). Using recursive feature elimination in random forest to account for correlated variables in high dimensional data. *BMC Genetics, 19*. doi:10.1186/s12863-018-0633-8

De Villiers, L. D., & Farrington, S. M. (2019). *The Role of Educational Factors in Identifying At-Risk Students in First Year Accounting.* Paper presented at the 31st Annual Conference of the Southern African Institute for Management Scientists (SAIMS).

DHET. (2020). *Department of Higher Education and Training Annual Report: 2019/2020.* DHET Retrieved from https://www.dhet.gov.za/Commissions%20Reports/DHET%20Annual%20Report%2019-20.pdf

DHET. (2021a). *Statistics on Post-School Education and Training in South Africa: 2019*. DHET Retrieved from https://www.dhet.gov.za/

DHET. (2021b). Post School Education and Training Monitor: 2021.

DHET. (2021c). *Dictionary of Terms and Concepts for Post-School Education and Training*.

Du, H. (2010). *Data Mining Techniques and Applications, An Introduction*.

Easterby-Smith, M., Thorpe, R., & Jackson, P. R. (2012). *Management research* (4th ed.). Los Angeles: Sage.

Er, E. (2012). Identifying at-risk students using machine learning techniques: A case study with IS 100. *International Journal of Machine Learning and Computing, 2*(4), 476.

Fan, J., Han, F., & Liu, H. (2014). Challenges of Big Data Analysis. *National science review, 1*(2), 293-314. doi:10.1093/nsr/nwt032

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (Second ed.). Canada: O'Reilly Media.

Goertzen, M. J. (2017). Introduction to quantitative research and data. *Library Technology Reports, 53*(4), 12-18.

Gold, L., & Albert, L. (2006). Graduation rates as a measure of college accountability. *American Academic, 2*(1), 89-106.

Government Technical Advisory Centre. (2016). *Post School Education and Training (PSET)*. GTAC Retrieved from https://www.gtac.gov.za/Pages/per_higheducation_pset.aspx

Haiyang, L., Wang, Z., Benachour, P., & Tubman, P. (2018). *A time series classification method for behaviour-based dropout prediction.* Paper presented at the 2018 IEEE 18th international conference on advanced learning technologies (ICALT).

Hassan, S.-U., Waheed, H., Aljohani, N., Ali, M., Ventura, S., & Herrera, F. (2019). Virtual Learning Environment to Predict Withdrawal by Leveraging Deep Learning. *International Journal of Intelligent Systems*. doi:10.1002/int.22129

Heuer, H., & Breiter, A. (2018). Student success prediction and the trade-off between big data and data minimization. *DeLFI -Die 16 E-Learning Fachtagung Informatik*.

Heydt, M. (2017). *Learning pandas* (Second ed.). Birmingham: Packt Publishing Ltd.

Hlosta, M., Zdrahal, Z., & Zendulka, J. (2017). *Ouroboros: early identification of at-risk students without models based on legacy data.* Paper presented at the Proceedings of the seventh international learning analytics & knowledge conference.

Hussain, M., Zhu, W., Zhang, W., & Abidi, S. M. R. (2018a). Student Engagement Predictions in an e-Learning System and Their Impact on Student Course Assessment Scores. *Computational Intelligence and Neuroscience, 2018*, 6347186. doi:10.1155/2018/6347186

Hussain, M., Zhu, W., Zhang, W., & Abidi, S. M. R. (2018b). Student engagement predictions in an e-learning system and their impact on student course assessment scores. *Computational intelligence neuroscience, 2018*.

Institute of Education Sciences. (2008). At Risk Students. Retrieved from https://eric.ed.gov/?qt=At+Risk+Students&ti=At+Risk+Students

Jesmeen, M. Z. H., Hossen, J., Sayeed, S., Ho, C. K., Tawsif, K., Rahman, M. A., & Hossain, M. (2018). A Survey on Cleaning Dirty Data Using Machine Learning Paradigm for Big Data Analytics. *Indonesian Journal of Electrical Engineering and Computer Science, 10*, 1234-1243. doi:10.11591/ijeecs.v10.i3.pp1234-1243

Jha, N. I., Ghergulescu, I., & Moldovan, A.-N. (2019). *OULAD MOOC Dropout and Result Prediction using Ensemble, Deep Learning and Regression Techniques.* Paper presented at the CSEDU (2).

Johannesson, P., & Perjons, E. (2014). *An Introduction to Design Science*. Switzerland: Springer.

Kabathova, J., & Drlík, M. (2021). Towards Predicting Student's Dropout in University Courses Using Different Machine Learning Techniques. *Applied Sciences, 11*, 3130. doi:10.3390/app11073130

Kakarla, R., Krishnan, S., & Alla, S. (2021). *Applied Data Science Using PySpark* (First ed.). USA: Apress.

Kotsiantis, S., Pierrakeas, C., & Pintelas, P. E. (2003). *Preventing student dropout in distance learning using machine learning techniques.* Paper presented at the International conference on knowledge-based and intelligent information and engineering systems.

Koutina, M., & Kermanidis, K. L. (2011). Predicting postgraduate students' performance using machine learning techniques. In *Artificial Intelligence Applications and Innovations* (pp. 159-168): Springer.

Kumar, R. (2011). *Research Methodology* (3rd ed.). London: SAGE Publications : SAGE Publications Ltd.

Kuzilek, J., Hlosta, M., & Zdrahal, Z. (2017). Open University Learning Analytics dataset. *Scientific Data, 4*(1), 170171. doi:10.1038/sdata.2017.171

Lakkaraju, H., Aguiar, E., Shan, C., Miller, D., Bhanpuri, N., Ghani, R., & Addison, K. L. (2015). A Machine Learning Framework to Identify Students at Risk of Adverse Academic Outcomes. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1909–1918): Association for Computing Machinery.

Larsen, M. R., Sommersel, H. B., & Larsen, M. S. (2013). *Evidence on dropout phenomena at universities*: Danish Clearinghouse for educational research Copenhagen.

Leavy, P. (2017). *Research design: Quantitative, qualitative, mixed methods, arts-based, and community-based participatory research approaches* (D. Laughton Ed. First ed.). New York, USA: The Guilford Press.

Lee, C.-Y., & Chen, B.-S. (2018). Mutually-exclusive-and-collectively-exhaustive feature selection scheme. *Applied Soft Computing, 68*, 961-971.

Leedy, P. D., & Ormrod, J. E. (2019). *Practical Research : Planning and Design* (Twelfth ed.). United Kingdom: Pearson Education Limited.

Levine, D. M., & Stephan, D. F. (2010). *Even you can learn statistics: A guide for everyone who has ever been afraid of statistics* (2nd ed.). New Jersey: FT Press.

Li, Y., Bellotti, T., & Adams, N. (2019). Issues using logistic regression with class imbalance, with a case study from credit risk modelling. *Foundations of Data Science, 1*(4), 389.

Livieris, I. E., Drakopoulou, K., & Pintelas, P. (2012). *Predicting students' performance using artificial neural networks.* Paper presented at the 8th PanHellenic Conference with International Participation Information and Communication Technologies in Education, University of Thessaly, Volos.

Lukman, A. F., Ayinde, K., & Ajiboye, A. S. (2017). Monte Carlo study of some classification-based ridge parameter estimators. *Journal of Modern Applied Statistical Methods, 16*(1), 24.

Luque, A., Carrasco, A., Martín, A., & de las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition, 91*, 216-231.

Lusigi, A. (2019). Higher Education, Technology, and Equity in Africa. *New Review of Information Networking, 24*(1), 1-16. doi:10.1080/13614576.2019.1608576

Lykourentzou, I., Giannoukos, I., Nikolopoulos, V., Mpardis, G., Loumos, V. J. C., & Education. (2009). Dropout prediction in e-learning courses through the combination of machine learning techniques. *53*(3), 950-965.

Mackenzie, N., & Knipe, S. (2006). Research dilemmas: Paradigms, methods and methodology. *Issues in educational research, 16*(2), 193-205.

Manafiazar, G., Riazi, M., Basarab, J. A., Li, C., Stothard, P., & Plastow, G. (2021). PSXV-29 Late-Breaking: Investigating use of machine learning algorithms to predict days in herd for commercial beef cattle. *Journal of Animal Science, 99*(Supplement_3), 381-381.

Maniraj, S., Saini, A., Ahmed, S., & Sarkar, S. (2019). Credit card fraud detection using machine learning and data science. *International Journal of Engineering Research and Technology, 8*(09), 1-7.

Mathews, L., & Hari, S. (2019). Learning from imbalanced data. In *Advanced Methodologies and Technologies in Network Architecture, Mobile Computing, and Data Analytics* (pp. 403-414): IGI Global.

McKinney, W. (2015). *Pandas-Powerful python data analysis toolkit*. In Vol. 1625. *Pandas—Powerful Python Data Analysis Toolkit* (pp. 297).

Meilman, P. W., Pattis, J. A., & Kraus-Zeilmann, D. (1994). Suicide attempts and threats on one college campus: Policy and practice. *Journal of American College Health, 42*(4), 147-154. doi:10.1080/07448481.1994.9939662

Milne, J., Jeffrey, L., Suddaby, G., & Higgins, A. (2012). Early identification of students at risk of failing. *Future Challenges, Sustainable Futures*, 657-661.

Ming, C., Xueshuai, Z., & Qian, J. (2019). *Challenges to Traditional Statistics in the Age of Big Data.* Paper presented at the 2019 International Conference on Arts, Management, Education and Innovation (ICAMEI 2019), Seoul, South Korea.

Misra, S., & Li, H. (2020). Noninvasive fracture characterization based on the classification of sonic wave travel times. In S. Misra, H. Li, & J. He (Eds.), *Machine Learning for Subsurface Characterization* (pp. 243-287). Oxford, United Kingdom.: Gulf Professional Publishing.

Mitchell, T. M. (1997). *Machine Learning*: McGraw-Hill Science/Engineering/Math, Inc.

Mofokeng, M. (2021). Higher Education budget vote: NSFAS gets more funds but shortage persist. Retrieved from https://insideeducation.co.za/2021/05/19/higher-education-budget-vote-nsfas-gets-more-funds-but-shortage-persist/

Murphy, K. P. (2012). *Machine Learning A Probabilistic Perspective*. Cambridge, Massachusetts: The MIT Press.

Najimi, A., Sharifirad, G., Amini, M. M., & Meftagh, S. D. (2013). Academic failure and students' viewpoint: The influence of individual, internal and external organizational factors. *Journal of education and health promotion, 2*, 22-22. doi:10.4103/2277-9531.112698

Nilashi, M., Ibrahim, O., Ahmadi, H., Shahmoradi, L., & Farahmand, M. (2018). A hybrid intelligent system for the prediction of Parkinson's Disease progression using machine learning techniques. *Biocybernetics Biomedical Engineering, 38*(1), 1-15.

Oates, B. J. (2006). *Researching Information Systems and Computing*. London: Sage.

Oladokun, V. O., Adebanjo, A. T., & Charles-Owaba, O. E. (2008). Predicting students' academic performance using artificial neural network: A case study of an engineering course. *The Pacific Journal of Science and Technology, 9*(1), 72-79.

Ouyang, F.-s., Guo, B.-l., Ouyang, L.-z., Liu, Z.-w., Lin, S.-j., Meng, W., . . . Yang, S.-m. (2019). Comparison between linear and nonlinear machine-learning algorithms for the classification of thyroid nodules. *European journal of radiology, 113*, 251-257.

Oyerinde, Y., & Chia, P. A. (2017). Predicting Students' Academic Performances – A Learning Analytics Approach using Multiple Linear Regression. *International Journal of Computer Applications, 157*, 37-44. doi:10.5120/ijca2017912671

Pandala, S. R. (2020). LazyPredict. 0.2.7. Retrieved from https://lazypredict.readthedocs.io/en/latest/readme.html

Pandey, M., & Taruna, S. (2018). An ensemble-based decision support system for the students' academic performance prediction. In *ICT Based Innovations* (pp. 163-169): Springer.

Patel, H., Shah, K., Sanghvi, D., & Manan, S. (2020). A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification. *Augmented Human Research, 5*(1), 16. doi:10.1007/s41133-020-00032-0

Pes, B. (2020). Learning from high-dimensional biomedical datasets: the issue of class imbalance. *IEEE Access, 8*, 13527-13540.

Phillips, D. C., & Burbules, N. C. (2000). *Postpositivism and educational research*. Lanham, MD, US: Rowman & Littlefield.

Picek, S., Heuser, A., Jovic, A., Bhasin, S., & Regazzoni, F. (2018). The Curse of Class Imbalance and Conflicting Metrics with Machine Learning for Side-channel Evaluations. *Transactions on Cryptographic Hardware and Embedded Systems, 2019*. doi:10.13154/tches.v2019.i1.209-237

Pillay, J. (2021). Suicidal behaviour among university students: a systematic review. *51*(1), 54-66. doi:10.1177/0081246321992177

Polikar, R. (2012). *Ensemble Machine Learning* (C. Zhang Ed. First ed.). New York, USA: Springer.

Polyzou, A., & Karypis, G. (2019). Feature Extraction for Next-Term Prediction of Poor Student Performance. *IEEE Transactions on Learning Technologies, 12*(2), 237-248. doi:10.1109/TLT.2019.2913358

Pouris, A., & Inglesi-Lotz, R. (2014). The contribution of higher education institutions to the South African economy. *South African Journal of Science, 110*, 01-07.

Prenkaj, B., Velardi, P., Stilo, G., Distante, D., & Faralli, S. (2020). A Survey of Machine Learning Approaches for Student Dropout Prediction in Online Courses. *53*(ACM Compututing Surveys), Article 57. doi:10.1145/3388792

Press, G. (2016). Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says.

Priya, S., & Uthra, R. A. (2021). Comprehensive analysis for class imbalance data with concept drift using ensemble based classification. *Journal of Ambient Intelligence Humanized Computing, 12*(5), 4943-4956.

Punch, K. F. (1998). *Introduction to social research: Quantitative and qualitative approaches*. London: SAGE Publications.

Qonde, G. F. (2021). *Briefing on the Revised Strategic Plan 2020 - 2025, Annual Performance Plan and Budget for 2021/22*. Retrieved from https://pmg.org.za/files/210526DHET_-_presentation.pptx

Radunzel, J. (2017). *Using Incoming Student Information to Identify Students At-Risk of Not Returning to Their Initial Institution in Year Two. ACT Research Report Series 2017-10*.

Rahman, M. H., & Islam, M. R. (2017). *Predict Student's Academic Performance and Evaluate the Impact of Different Attributes on the Performance Using Data Mining Techniques.* Paper presented at the 2017 2nd International Conference on Electrical & Electronic Engineering (ICEEE).

Rajandran, K., Hee, T., Kanawarthy, S., Soon, L., Kamaludin, H., & Khezrimotlagh, D. (2014). Factors Affecting First Year Undergraduate Students Academic Performance.

Raschka, S., & Mirjalili, V. (2017). *Python Machine Learning: Machine Learning and Deep Learning with Python Scikit-Learn, and TensorFlow* (M. Sangwan Ed. Second ed.). Birmingham, UK: Packt Publishing.

Rizvi, S., Rienties, B., & Khoja, S. A. (2019). The role of demographics in online learning; A decision tree based approach. *Computers Education, 137*, 32-47.

Rovira, S., Puertas, E., & Igual, L. (2017). Data-driven system to predict academic grades and dropout. *PLoS one, 12*(2), e0171207.

Rowntree, D. (2004). *Statistics without tears : a primer for non-mathematicians*. Boston: Allyn and Bacon.

Rubin, A., & Babbie, E. R. (2016). *Essential research methods for social work* (9th ed.). Boston, US: Cengage Learning.

Sahin, E. K. (2020). Assessing the predictive capability of ensemble tree methods for landslide susceptibility mapping using XGBoost, gradient boosting machine, and random forest. *SN Applied Sciences, 2*(7), 1308. doi:10.1007/s42452-020-3060-1

Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development, 44*(3.3), 210-229. doi:10.1147/rd.441.0206

Sarka, D. (2021). Descriptive statistics. In *Advanced Analytics with Transact-SQL* (pp. 3-29): Springer.

Sarkar, S., Khatedi, N., Pramanik, A., & Maiti, J. (2020). An ensemble learning-based undersampling technique for handling class-imbalance problem. In *Proceedings of ICETIT 2019* (pp. 586-595): Springer.

Shah, A. (2016). Machine Learning Vs Statistics. Retrieved from https://www.kdnuggets.com/2016/11/machine-learning-vs-statistics.html

Shahbaz, M. S., Rasi, R. Z. R., Ahmad, M. F. B., & Management. (2019). A novel classification of supply chain risks: Scale development and validation. *Journal of Industrial Engineering, 12*(1), 201-218.

Shaw, E., & Mattern, K. (2013). Examining Student Under- and Overperformance in College to Identify Risk of Attrition. *Educational Assessment, 18*, 251-268. doi:10.1080/10627197.2013.846676

Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access, 7*, 53040-53065.

Siri, A. (2015). Predicting students' dropout at university using artificial neural networks. *Italian Journal of Sociology of Education, 7*(2).

Smith, M., Therry, L., & Whale, J. (2012). Developing a Model for Identifying Students at Risk of Failure in a First Year Accounting Unit. *Higher Education Studies, 2*, 91-102. doi:10.5539/hes.v2n4p91

Solares, J. R. A., Wei, H.-L., & Billings, S. A. (2019). A novel logistic-NARX model as a classifier for dynamic binary classification. *Neural Computing Applications, 31*(1), 11-25.

Soobramoney, R., & Singh, A. (2019, 6-8 March 2019). *Identifying Students At-Risk with an Ensemble of Machine Learning Algorithms.* Paper presented at the 2019 Conference on Information Communications Technology and Society (ICTAS).

Spence, S. (2012). *Operation Student Success: Griffith's Student Retention Strategy 2012 – 2014*. Retrieved from Griffith University, Australia: https://intranet.secure.griffith.edu.au/__data/assets/pdf_file/0021/37470/Student_Retention_Strategy-2012-2014.pdf

Stellenbosch University. (2017). *Stellenbosch University supports 8% income increase for 2017*. Retrieved from http://www.sun.ac.za/english/Documents/Student%20fees/Stellenbosch%20University%20supports%20income%20increase%20for%202017.pdf

Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International journal of pattern recognition artificial intelligence, 23*(04), 687-719.

The Glossary of Educational Reform. (2013). At-Risk. Retrieved from https://www.edglossary.org/at-risk/

Thompson, R., Li, J., & Shulruf, B. (2019). Struggling with strugglers: using data from selection tools for early identification of medical students at risk of failure. *BMC Medical Education, 19*(1), 415. doi:10.1186/s12909-019-1860-z

Tinto, V. (1975). Dropout from higher education: A theoretical Synthesis of Recent Research. *Review of Educational Research, 45 (1)*, 89-125.

Tinto, V. (1987). *Leaving college: Rethinking the causes and cures of student attrition*: ERIC.

Tinto, V. (2005). *College Student Retention: Formula for Student Success* (A. Seidman Ed.). Westport, CT: Praeger Publishers.

Tinto, V. (2012). Enhancing student success: Taking the classroom success seriously. *Student Success, 3*(1), 1.

Urrutia-Aguilar, M., Fuentes-Garcia, R., Martinez, D., Beck, E., Ortiz, S., & Guevara-Guzmán, R. (2016). Logistic Regression Model for the Academic Performance of First-Year Medical Students in the Biomedical Area. *Creative Education, 07*, 2202-2211. doi:10.4236/ce.2016.715217

Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., & Anderla, A. (2019). *Credit card fraud detection-machine learning methods.* Paper presented at the 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH).

Veerasamy, A. K., D'Souza, D., Apiola, M.-V., Laakso, M.-J., & Salakoski, T. (2020). *Using early assessment performance as early warning signs to identify at-risk students in programming courses.* Paper presented at the 2020 IEEE Frontiers in Education Conference (FIE).

Wasif, M., Waheed, H., Aljohani, N. R., & Hassan, S.-U. (2019). Understanding student learning behavior and predicting their performance. In *Cognitive Computing in Technology-Enhanced Learning* (pp. 1-28): IGI Global.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation, 1*(1), 67-82.

Xing, W., Chen, X., Stein, J., & Marcinkowski, M. (2016). Temporal predication of dropouts in MOOCs: Reaching the low hanging fruit through stacking generalization. *Computers in Human Behavior, 58*(May 2016), 119-129. doi:https://doi.org/10.1016/j.chb.2015.12.007

Xu, J.-L., Esquerre, C., & Sun, D.-W. (2018). Methods for performing dimensionality reduction in hyperspectral image classification. *Journal of Near Infrared Spectroscopy, 26*(1), 61-75.

Yee, O. S., Sagadevan, S., & Malim, N. H. A. H. (2018). Credit card fraud detection using machine learning as data mining technique. *Journal of Telecommunication, Electronic Computer Engineering, 10*(1-4), 23-27.

Younge, S. L., Oetting, E. R., & Deffenbacher, J. L. (1996). Correlations among Maternal Rejection, Dropping Out of School, and Drug Use in Adolescents: A Pilot Study. *J Clin Psychol, 52*(1), 96-102.

Zamayirha, P. (2018). Mental health, fees, trauma. Why SA's students commit suicide. Retrieved from https://www.news24.com/citypress/news/mental-health-fees-trauma-why-sas-students-commit-suicide-20180930

Zheng, A. (2015). *Evaluating machine learning models : a beginner's guide to key concepts and pitfalls* (First ed.). CA, United States of America: O'Reilly Media.

Zhou, Z.-H. (2012). *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL: Chapman and Hall/CRC Press.

Appendix A: Confusion Matrix

The confusion matrix for all 25 MLAs for each of the 40 weeks is shown in Table A1 (split owing to the size of the tables).

Table A1: Confusion matrix for all 40 weeks.

ML Algorithm	Wk0	Wk1	Wk2	Wk3	Wk4	Wk5	Wk6	Wk7	Wk8	Wk9	Wk10	Wk11	Wk12	Wk13	Wk14	Wk15	Wk16	Wk17	Wk18	Wk19
AdaBoostClassifier	[[1533 961] [899 1822]]	[[1560 934] [850 1871]]	[[1535 959] [777 1944]]	[[1557 937] [747 1974]]	[[1603 891] [749 1972]]	[[1619 875] [720 2001]]	[[1655 839] [673 2048]]	[[1693 801] [608 2113]]	[[1706 788] [601 2120]]	[[1716 778] [575 2146]]	[[1749 745] [575 2146]]	[[1775 719] [569 2152]]	[[1774 720] [584 2137]]	[[1822 672] [535 2186]]	[[1836 658] [517 2204]]	[[1863 631] [506 2215]]	[[1898 596] [496 2225]]	[[1914 580] [490 2231]]	[[1949 545] [461 2260]]	[[1954 540] [431 2290]]
BaggingClassifier	[[1524 970] [1153 1568]]	[[1599 895] [1139 1582]]	[[1609 885] [1047 1674]]	[[1612 882] [1044 1677]]	[[1666 828] [1000 1721]]	[[1700 794] [963 1758]]	[[1732 762] [908 1813]]	[[1710 784] [823 1898]]	[[1746 748] [786 1935]]	[[1741 753] [799 1922]]	[[1728 766] [755 1966]]	[[1727 767] [731 1990]]	[[1727 767] [720 2001]]	[[1777 717] [661 2060]]	[[1788 706] [644 2077]]	[[1823 671] [570 2151]]	[[1844 650] [585 2136]]	[[1867 627] [515 2206]]	[[1907 587] [486 2235]]	[[1903 591] [458 2263]]
BernoulliNB	[[1457 1037] [926 1795]]	[[1514 980] [929 1792]]	[[1632 862] [982 1739]]	[[1684 810] [1033 1688]]	[[1723 771] [1107 1614]]	[[1764 730] [1153 1568]]	[[1830 664] [1188 1533]]	[[1860 634] [1200 1521]]	[[1861 633] [1203 1518]]	[[1890 604] [1182 1539]]	[[1923 571] [1183 1538]]	[[1934 560] [1201 1520]]	[[1949 545] [1196 1525]]	[[1968 526] [1189 1532]]	[[1983 511] [1174 1547]]	[[1989 505] [1172 1549]]	[[2016 478] [1149 1572]]	[[2037 457] [1123 1598]]	[[2064 430] [1095 1626]]	[[2065 429] [1084 1637]]
CalibratedClassifierCV	[[1470 1024] [924 1797]]	[[1520 974] [907 1814]]	[[1572 922] [882 1839]]	[[1574 920] [867 1854]]	[[1592 902] [856 1865]]	[[1624 870] [835 1886]]	[[1653 841] [827 1894]]	[[1682 812] [825 1896]]	[[1675 819] [813 1908]]	[[1723 771] [812 1909]]	[[1717 777] [777 1944]]	[[1729 765] [777 1944]]	[[1749 745] [777 1944]]	[[1776 718] [769 1952]]	[[1805 689] [772 1949]]	[[1837 657] [752 1969]]	[[1860 634] [743 1978]]	[[1884 610] [745 1976]]	[[1934 560] [775 1946]]	[[1912 582] [704 2017]]
DecisionTreeClassifier	[[1403 1091] [1117 1604]]	[[1371 1123] [1123 1598]]	[[1405 1089] [1046 1675]]	[[1463 1031] [1057 1664]]	[[1499 995] [1070 1651]]	[[1530 964] [1037 1684]]	[[1590 904] [951 1770]]	[[1563 931] [909 1812]]	[[1589 905] [934 1787]]	[[1593 901] [954 1767]]	[[1648 846] [836 1885]]	[[1651 843] [874 1847]]	[[1652 842] [847 1874]]	[[1686 808] [857 1864]]	[[1718 776] [788 1933]]	[[1755 739] [773 1948]]	[[1784 710] [769 1952]]	[[1808 686] [753 1968]]	[[1834 660] [722 1999]]	[[1865 629] [663 2058]]
DummyClassifier	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]
ExtraTreeClassifier	[[1415 1079] [1183 1538]]	[[1371 1123] [1150 1571]]	[[1392 1102] [1141 1580]]	[[1443 1051] [1126 1595]]	[[1432 1062] [1084 1637]]	[[1468 1026] [1070 1651]]	[[1480 1014] [1022 1699]]	[[1576 918] [1043 1678]]	[[1509 985] [993 1728]]	[[1527 967] [921 1800]]	[[1572 922] [927 1794]]	[[1597 897] [953 1768]]	[[1601 893] [940 1781]]	[[1601 893] [900 1821]]	[[1642 852] [853 1868]]	[[1713 781] [858 1863]]	[[1711 783] [832 1889]]	[[1719 775] [810 1911]]	[[1724 770] [810 1911]]	[[1820 674] [760 1961]]
ExtraTreesClassifier	[[1405 1089] [1117 1604]]	[[1443 1051] [998 1723]]	[[1505 989] [917 1804]]	[[1530 964] [865 1856]]	[[1543 951] [799 1922]]	[[1580 914] [739 1982]]	[[1605 889] [690 2031]]	[[1609 885] [631 2090]]	[[1606 888] [588 2133]]	[[1648 846] [577 2144]]	[[1653 841] [501 2220]]	[[1675 819] [525 2196]]	[[1674 820] [482 2239]]	[[1683 811] [436 2285]]	[[1722 772] [403 2318]]	[[1750 744] [371 2350]]	[[1735 759] [362 2359]]	[[1764 730] [330 2391]]	[[1794 700] [296 2425]]	[[1809 685] [283 2438]]
GaussianNB	[[1723 771] [1380 1341]]	[[1798 696] [1417 1304]]	[[1866 628] [1478 1243]]	[[1932 562] [1499 1222]]	[[1991 503] [1587 1134]]	[[2042 452] [1628 1093]]	[[2073 421] [1641 1080]]	[[2116 378] [1681 1040]]	[[2135 359] [1740 981]]	[[2163 331] [1773 948]]	[[2150 344] [1783 938]]	[[2162 332] [1795 926]]	[[2168 326] [1798 923]]	[[2186 308] [1786 935]]	[[2187 307] [1757 964]]	[[2189 305] [1752 969]]	[[2203 291] [1717 1004]]	[[2211 283] [1688 1033]]	[[2204 290] [1650 1071]]	[[2204 290] [1625 1096]]
KNeighborsClassifier	[[1437 1057] [1119 1602]]	[[1480 1014] [1096 1625]]	[[1532 962] [1097 1624]]	[[1525 969] [1065 1656]]	[[1550 944] [1050 1671]]	[[1579 915] [1029 1692]]	[[1630 864] [1003 1718]]	[[1654 840] [990 1731]]	[[1653 841] [993 1728]]	[[1688 806] [1014 1707]]	[[1689 805] [992 1729]]	[[1727 767] [1013 1708]]	[[1735 759] [1021 1700]]	[[1760 734] [1008 1713]]	[[1780 714] [1001 1720]]	[[1800 694] [989 1732]]	[[1837 657] [993 1728]]	[[1869 625] [995 1726]]	[[1883 611] [989 1732]]	[[1905 589] [981 1740]]
LGBMClassifier	[[1510 984] [887 1834]]	[[1528 966] [830 1891]]	[[1515 979] [748 1973]]	[[1516 978] [726 1995]]	[[1556 938] [668 2053]]	[[1573 921] [639 2082]]	[[1624 870] [588 2133]]	[[1652 842] [527 2194]]	[[1628 866] [502 2219]]	[[1651 843] [474 2247]]	[[1652 842] [425 2296]]	[[1675 819] [435 2286]]	[[1689 805] [430 2291]]	[[1698 796] [394 2327]]	[[1726 768] [376 2345]]	[[1755 739] [339 2382]]	[[1783 711] [334 2387]]	[[1816 678] [301 2420]]	[[1835 659] [288 2433]]	[[1837 657] [257 2464]]
LinearDiscriminantAnalysis	[[1472 1022] [939 1782]]	[[1504 990] [925 1796]]	[[1566 928] [900 1821]]	[[1568 926] [880 1841]]	[[1596 898] [878 1843]]	[[1622 872] [864 1857]]	[[1656 838] [861 1860]]	[[1684 810] [862 1859]]	[[1684 810] [848 1873]]	[[1700 794] [853 1868]]	[[1697 797] [847 1874]]	[[1722 772] [843 1878]]	[[1729 765] [855 1866]]	[[1758 736] [835 1886]]	[[1785 709] [843 1878]]	[[1795 699] [834 1887]]	[[1816 678] [831 1890]]	[[1838 656] [836 1885]]	[[1858 636] [833 1888]]	[[1883 611] [787 1934]]
LinearSVC	[[1480 1014] [933 1788]]	[[1533 961] [915 1806]]	[[1589 905] [900 1821]]	[[1584 910] [889 1832]]	[[1607 887] [872 1849]]	[[1650 844] [856 1865]]	[[1675 819] [845 1876]]	[[1709 785] [849 1872]]	[[1710 784] [833 1888]]	[[1748 746] [833 1888]]	[[1735 759] [797 1924]]	[[1763 731] [807 1914]]	[[1768 726] [793 1928]]	[[1779 715] [784 1937]]	[[1831 663] [793 1928]]	[[1878 616] [788 1933]]	[[1890 604] [781 1940]]	[[1912 582] [782 1939]]	[[1943 551] [799 1922]]	[[1966 528] [737 1984]]
LogisticRegression	[[1492 1002] [928 1793]]	[[1521 973] [909 1812]]	[[1589 905] [888 1833]]	[[1600 894] [876 1845]]	[[1618 876] [856 1865]]	[[1642 852] [840 1881]]	[[1666 828] [827 1894]]	[[1722 772] [829 1892]]	[[1731 763] [808 1913]]	[[1753 741] [805 1916]]	[[1747 747] [782 1939]]	[[1769 725] [784 1937]]	[[1776 718] [777 1944]]	[[1804 690] [768 1953]]	[[1839 655] [771 1950]]	[[1869 625] [741 1980]]	[[1903 591] [755 1966]]	[[1931 563] [755 1966]]	[[1965 529] [757 1964]]	[[1985 509] [714 2007]]
NearestCentroid	[[1562 932] [1072 1649]]	[[1637 857] [1067 1654]]	[[1734 760] [1101 1620]]	[[1790 704] [1151 1570]]	[[1857 637] [1233 1488]]	[[1922 572] [1302 1419]]	[[1966 528] [1341 1380]]	[[1999 495] [1376 1345]]	[[2010 484] [1391 1330]]	[[2052 442] [1406 1315]]	[[2062 432] [1423 1298]]	[[2079 415] [1455 1266]]	[[2089 405] [1457 1264]]	[[2102 392] [1442 1279]]	[[2112 382] [1437 1284]]	[[2127 367] [1440 1281]]	[[2141 353] [1431 1290]]	[[2151 343] [1429 1292]]	[[2164 330] [1426 1295]]	[[2176 318] [1417 1304]]
NuSVC	[[1456 1038] [1109 1612]]	[[1460 1034] [1056 1665]]	[[1540 954] [1036 1685]]	[[1537 957] [996 1725]]	[[1559 935] [966 1755]]	[[1599 895] [938 1783]]	[[1622 872] [884 1837]]	[[1650 844] [827 1894]]	[[1645 849] [779 1942]]	[[1673 821] [747 1974]]	[[1675 819] [667 2054]]	[[1714 780] [695 2026]]	[[1733 761] [693 2028]]	[[1759 735] [647 2074]]	[[1816 678] [596 2125]]	[[1859 635] [555 2166]]	[[1874 620] [545 2176]]	[[1888 606] [535 2186]]	[[1924 570] [534 2187]]	[[1966 528] [518 2203]]
PassiveAggressiveClassifier	[[1083 1411] [920 1801]]	[[1238 1256] [1066 1655]]	[[1234 1260] [876 1845]]	[[1273 1221] [860 1861]]	[[1278 1216] [836 1885]]	[[1276 1218] [833 1888]]	[[1445 1049] [1166 1555]]	[[1536 958] [1093 1628]]	[[1663 831] [1150 1571]]	[[1381 1113] [817 1904]]	[[1544 950] [998 1723]]	[[1564 930] [973 1748]]	[[1581 913] [961 1760]]	[[1631 863] [968 1753]]	[[1779 715] [997 1724]]	[[1446 1048] [923 1798]]	[[1433 1061] [892 1829]]	[[1433 1061] [898 1823]]	[[1410 1084] [859 1862]]	[[1429 1065] [885 1836]]
Perceptron	[[1526 968] [1540 1181]]	[[1358 1136] [1324 1397]]	[[1356 1138] [1228 1493]]	[[1786 708] [1370 1351]]	[[1294 1200] [1000 1721]]	[[1277 1217] [820 1901]]	[[1352 1142] [1002 1719]]	[[1450 1044] [845 1876]]	[[1519 975] [1134 1587]]	[[1303 1191] [782 1939]]	[[1728 766] [1163 1558]]	[[1413 1081] [790 1931]]	[[1423 1071] [962 1759]]	[[1563 931] [877 1844]]	[[1831 663] [1024 1697]]	[[1686 808] [1007 1714]]	[[1883 611] [955 1766]]	[[1878 616] [961 1760]]	[[1792 702] [828 1893]]	[[2086 408] [1087 1634]]
QuadraticDiscriminantAnalysis	[[1567 927] [1281 1440]]	[[1704 790] [1518 1203]]	[[1616 878] [1283 1438]]	[[1677 817] [1430 1291]]	[[1000 1494] [811 1910]]	[[1587 907] [1319 1402]]	[[1891 603] [1508 1213]]	[[1986 508] [1750 971]]	[[1852 642] [1535 1186]]	[[2055 439] [1692 1029]]	[[1880 614] [1485 1236]]	[[1893 601] [1503 1218]]	[[1794 700] [1376 1345]]	[[1947 547] [1357 1364]]	[[2071 423] [1489 1232]]	[[2105 389] [1541 1180]]	[[1889 605] [1332 1389]]	[[2176 318] [1611 1110]]	[[2149 345] [1521 1200]]	[[2101 393] [1458 1263]]
RandomForestClassifier	[[1427 1067] [965 1756]]	[[1502 992] [881 1840]]	[[1524 970] [820 1901]]	[[1534 960] [766 1955]]	[[1559 935] [715 2006]]	[[1568 926] [678 2043]]	[[1604 890] [611 2110]]	[[1603 891] [555 2166]]	[[1600 894] [505 2216]]	[[1621 873] [469 2252]]	[[1611 883] [434 2287]]	[[1617 877] [430 2291]]	[[1632 862] [399 2322]]	[[1658 836] [383 2338]]	[[1685 809] [373 2348]]	[[1721 773] [324 2397]]	[[1743 751] [329 2392]]	[[1762 732] [277 2444]]	[[1779 715] [264 2457]]	[[1800 694] [218 2503]]
RidgeClassifier	[[1472 1022] [939 1782]]	[[1504 990] [925 1796]]	[[1566 928] [900 1821]]	[[1568 926] [880 1841]]	[[1596 898] [877 1844]]	[[1622 872] [864 1857]]	[[1656 838] [861 1860]]	[[1684 810] [862 1859]]	[[1684 810] [848 1873]]	[[1700 794] [853 1868]]	[[1697 797] [847 1874]]	[[1722 772] [843 1878]]	[[1729 765] [855 1866]]	[[1758 736] [835 1886]]	[[1785 709] [843 1878]]	[[1795 699] [833 1888]]	[[1816 678] [831 1890]]	[[1838 656] [835 1886]]	[[1858 636] [833 1888]]	[[1883 611] [787 1934]]
RidgeClassifierCV	[[1472 1022] [939 1782]]	[[1504 990] [925 1796]]	[[1566 928] [901 1820]]	[[1568 926] [880 1841]]	[[1595 899] [878 1843]]	[[1622 872] [864 1857]]	[[1656 838] [861 1860]]	[[1684 810] [862 1859]]	[[1684 810] [847 1874]]	[[1700 794] [853 1868]]	[[1697 797] [847 1874]]	[[1722 772] [843 1878]]	[[1729 765] [855 1866]]	[[1758 736] [835 1886]]	[[1785 709] [843 1878]]	[[1795 699] [834 1887]]	[[1816 678] [831 1890]]	[[1838 656] [836 1885]]	[[1858 636] [833 1888]]	[[1883 611] [787 1934]]
SGDClassifier	[[1366 1128] [953 1768]]	[[1572 922] [1080 1641]]	[[1605 889] [1070 1651]]	[[1515 979] [957 1764]]	[[1492 1002] [899 1822]]	[[1613 881] [918 1803]]	[[1776 718] [1030 1691]]	[[1750 744] [918 1803]]	[[1488 1006] [698 2023]]	[[1530 964] [701 2020]]	[[1491 1003] [613 2108]]	[[1511 983] [628 2093]]	[[1504 990] [625 2096]]	[[1578 916] [632 2089]]	[[1556 938] [594 2127]]	[[1790 704] [700 2021]]	[[1698 796] [647 2074]]	[[1686 808] [633 2088]]	[[2118 376] [938 1783]]	[[2017 477] [791 1930]]
SVC	[[1461 1033] [910 1811]]	[[1479 1015] [835 1886]]	[[1527 967] [796 1925]]	[[1506 988] [760 1961]]	[[1515 979] [721 2000]]	[[1543 951] [697 2024]]	[[1596 898] [678 2043]]	[[1643 851] [654 2067]]	[[1619 875] [633 2088]]	[[1675 819] [651 2070]]	[[1644 850] [580 2141]]	[[1686 808] [581 2140]]	[[1709 785] [585 2136]]	[[1751 743] [579 2142]]	[[1805 689] [568 2153]]	[[1842 652] [544 2177]]	[[1867 627] [542 2179]]	[[1882 612] [534 2187]]	[[1929 565] [529 2192]]	[[1963 531] [515 2206]]
XGBClassifier	[[1490 1004] [920 1801]]	[[1524 970] [903 1818]]	[[1512 982] [816 1905]]	[[1531 963] [786 1935]]	[[1559 935] [757 1964]]	[[1577 917] [711 2010]]	[[1639 855] [644 2077]]	[[1639 855] [621 2100]]	[[1632 862] [561 2160]]	[[1660 834] [560 2161]]	[[1682 812] [547 2174]]	[[1715 779] [537 2184]]	[[1686 808] [501 2220]]	[[1718 776] [457 2264]]	[[1723 771] [424 2297]]	[[1774 720] [429 2292]]	[[1780 714] [384 2337]]	[[1822 672] [359 2362]]	[[1852 642] [348 2373]]	[[1881 613] [314 2407]]

ML Algorithm	Wk20	Wk21	Wk22	Wk23	Wk24	Wk25	Wk26	Wk27	Wk28	Wk29	Wk30	Wk31	Wk32	Wk33	Wk34	Wk35	Wk36	Wk37	Wk38	Wk39
AdaBoostClassifier	[[1985 509] [426 2295]]	[[2006 488] [430 2291]]	[[2020 474] [402 2319]]	[[2058 436] [391 2330]]	[[2084 410] [378 2343]]	[[2098 396] [367 2354]]	[[2117 377] [371 2350]]	[[2117 377] [364 2357]]	[[2128 366] [343 2378]]	[[2157 337] [330 2391]]	[[2164 330] [322 2399]]	[[2182 312] [312 2409]]	[[2201 293] [311 2410]]	[[2224 270] [286 2435]]	[[2228 266] [281 2440]]	[[2232 262] [287 2434]]	[[2230 264] [283 2438]]	[[2230 264] [283 2438]]	[[2230 264] [283 2438]]	[[2230 264] [283 2438]]
BaggingClassifier	[[1910 584] [453 2268]]	[[1953 541] [401 2320]]	[[1957 537] [417 2304]]	[[1988 506] [332 2389]]	[[2009 485] [322 2399]]	[[2024 470] [327 2394]]	[[2030 464] [312 2409]]	[[2046 448] [304 2417]]	[[2068 426] [256 2465]]	[[2071 423] [239 2482]]	[[2093 401] [206 2515]]	[[2101 393] [199 2522]]	[[2126 368] [198 2523]]	[[2139 355] [186 2535]]	[[2154 340] [147 2574]]	[[2164 330] [148 2573]]	[[2155 339] [156 2565]]	[[2154 340] [157 2564]]	[[2155 339] [157 2564]]	[[2158 336] [156 2565]]
BernoulliNB	[[2072 422] [1059 1662]]	[[2094 400] [1048 1673]]	[[2112 382] [1038 1683]]	[[2119 375] [1016 1705]]	[[2118 376] [984 1737]]	[[2129 365] [977 1744]]	[[2144 350] [978 1743]]	[[2152 342] [975 1746]]	[[2161 333] [952 1769]]	[[2171 323] [924 1797]]	[[2175 319] [889 1832]]	[[2190 304] [882 1839]]	[[2193 301] [876 1845]]	[[2199 295] [859 1862]]	[[2202 292] [833 1888]]	[[2198 296] [816 1905]]	[[2200 294] [804 1917]]	[[2204 290] [797 1924]]	[[2206 288] [797 1924]]	[[2206 288] [794 1927]]
CalibratedClassifierCV	[[1965 529] [702 2019]]	[[1972 522] [684 2037]]	[[2005 489] [654 2067]]	[[2037 457] [640 2081]]	[[2045 449] [637 2084]]	[[2073 421] [631 2090]]	[[2096 398] [631 2090]]	[[2114 380] [620 2101]]	[[2129 365] [599 2122]]	[[2162 332] [578 2143]]	[[2159 335] [565 2156]]	[[2171 323] [551 2170]]	[[2184 310] [537 2184]]	[[2209 285] [535 2186]]	[[2225 269] [524 2197]]	[[2226 268] [491 2230]]	[[2233 261] [488 2233]]	[[2235 259] [491 2230]]	[[2242 252] [504 2217]]	[[2241 253] [510 2211]]
DecisionTreeClassifier	[[1866 628] [650 2071]]	[[1908 586] [662 2059]]	[[1905 589] [622 2099]]	[[1974 520] [552 2169]]	[[1968 526] [553 2168]]	[[2006 488] [563 2158]]	[[2031 463] [532 2189]]	[[2023 471] [478 2243]]	[[2070 424] [502 2219]]	[[2082 412] [454 2267]]	[[2080 414] [451 2270]]	[[2086 408] [503 2218]]	[[2121 373] [425 2296]]	[[2119 375] [418 2303]]	[[2139 355] [383 2338]]	[[2144 350] [363 2358]]	[[2147 347] [374 2347]]	[[2140 354] [386 2335]]	[[2150 344] [372 2349]]	[[2140 354] [378 2343]]
DummyClassifier	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]	[[1186 1308] [1296 1425]]
ExtraTreeClassifier	[[1771 723] [721 2000]]	[[1842 652] [677 2044]]	[[1847 647] [661 2060]]	[[1880 614] [625 2096]]	[[1907 587] [657 2064]]	[[1898 596] [618 2103]]	[[1923 571] [652 2069]]	[[1941 553] [602 2119]]	[[1951 543] [580 2141]]	[[1960 534] [554 2167]]	[[1972 522] [575 2146]]	[[1974 520] [498 2223]]	[[2055 439] [529 2192]]	[[2040 454] [519 2202]]	[[2065 429] [475 2246]]	[[2089 405] [464 2257]]	[[2092 402] [471 2250]]	[[2094 400] [434 2287]]	[[2029 465] [492 2229]]	[[2035 459] [488 2233]]
ExtraTreesClassifier	[[1819 675] [253 2468]]	[[1852 642] [237 2484]]	[[1867 627] [208 2513]]	[[1904 590] [182 2539]]	[[1907 587] [158 2563]]	[[1937 557] [140 2581]]	[[1938 556] [130 2591]]	[[1957 537] [123 2598]]	[[1964 530] [104 2617]]	[[1991 503] [93 2628]]	[[2010 484] [89 2632]]	[[2028 466] [77 2644]]	[[2039 455] [70 2651]]	[[2060 434] [65 2656]]	[[2079 415] [55 2666]]	[[2081 413] [40 2681]]	[[2085 409] [42 2679]]	[[2087 407] [44 2677]]	[[2082 412] [45 2676]]	[[2084 410] [50 2671]]
GaussianNB	[[2202 292] [1603 1118]]	[[2205 289] [1587 1134]]	[[2212 282] [1562 1159]]	[[2222 272] [1527 1194]]	[[2229 265] [1516 1205]]	[[2236 258] [1505 1216]]	[[2255 239] [1487 1234]]	[[2265 229] [1485 1236]]	[[2273 221] [1470 1251]]	[[2280 214] [1460 1261]]	[[2291 203] [1451 1270]]	[[2295 199] [1444 1277]]	[[2296 198] [1445 1276]]	[[2292 202] [1426 1295]]	[[2297 197] [1410 1311]]	[[2295 199] [1414 1307]]	[[2296 198] [1423 1298]]	[[2298 196] [1397 1324]]	[[2304 190] [1409 1312]]	[[2304 190] [1418 1303]]
KNeighborsClassifier	[[1932 562] [972 1749]]	[[1936 558] [956 1765]]	[[1955 539] [929 1792]]	[[1982 512] [927 1794]]	[[2000 494] [922 1799]]	[[2012 482] [924 1797]]	[[2027 467] [927 1794]]	[[2037 457] [937 1784]]	[[2059 435] [931 1790]]	[[2069 425] [917 1804]]	[[2085 409] [895 1826]]	[[2097 397] [891 1830]]	[[2105 389] [886 1835]]	[[2118 376] [881 1840]]	[[2125 369] [861 1860]]	[[2125 369] [860 1861]]	[[2129 365] [855 1866]]	[[2129 365] [866 1855]]	[[2134 360] [872 1849]]	[[2135 359] [879 1842]]
LGBMClassifier	[[1885 609] [251 2470]]	[[1912 582] [237 2484]]	[[1937 557] [198 2523]]	[[1946 548] [197 2524]]	[[1968 526] [171 2550]]	[[1983 511] [136 2585]]	[[1999 495] [153 2568]]	[[2015 479] [162 2559]]	[[2035 459] [141 2580]]	[[2062 432] [127 2594]]	[[2095 399] [107 2614]]	[[2108 386] [109 2612]]	[[2133 361] [111 2610]]	[[2142 352] [110 2611]]	[[2153 341] [97 2624]]	[[2161 333] [81 2640]]	[[2164 330] [95 2626]]	[[2159 335] [91 2630]]	[[2159 335] [86 2635]]	[[2157 337] [88 2633]]
LinearDiscriminantAnalysis	[[1905 589] [800 1921]]	[[1921 573] [780 1941]]	[[1942 552] [779 1942]]	[[1951 543] [771 1950]]	[[1955 539] [764 1957]]	[[1954 540] [757 1964]]	[[1960 534] [751 1970]]	[[1972 522] [754 1967]]	[[1990 504] [740 1981]]	[[1997 497] [726 1995]]	[[2021 473] [711 2010]]	[[2018 476] [712 2009]]	[[2022 472] [704 2017]]	[[2031 463] [675 2046]]	[[2055 439] [639 2082]]	[[2056 438] [635 2086]]	[[2052 442] [627 2094]]	[[2056 438] [622 2099]]	[[2056 438] [620 2101]]	[[2058 436] [625 2096]]
LinearSVC	[[2003 491] [755 1966]]	[[2010 484] [721 2000]]	[[2047 447] [700 2021]]	[[2062 432] [686 2035]]	[[2085 409] [678 2043]]	[[2102 392] [657 2064]]	[[2113 381] [646 2075]]	[[2141 353] [656 2065]]	[[2159 335] [636 2085]]	[[2174 320] [613 2108]]	[[2178 316] [585 2136]]	[[2195 299] [571 2150]]	[[2197 297] [556 2165]]	[[2200 294] [539 2182]]	[[2185 309] [524 2197]]	[[2222 272] [485 2236]]	[[2219 275] [486 2235]]	[[2223 271] [487 2234]]	[[2219 275] [489 2232]]	[[2225 269] [491 2230]]
LogisticRegression	[[2014 480] [711 2010]]	[[2019 475] [687 2034]]	[[2061 433] [650 2071]]	[[2081 413] [638 2083]]	[[2110 384] [611 2110]]	[[2123 371] [594 2127]]	[[2133 361] [606 2115]]	[[2139 355] [602 2119]]	[[2164 330] [571 2150]]	[[2185 309] [535 2186]]	[[2197 297] [510 2211]]	[[2205 289] [513 2208]]	[[2209 285] [503 2218]]	[[2212 282] [484 2237]]	[[2223 271] [449 2272]]	[[2222 272] [426 2295]]	[[2226 268] [437 2284]]	[[2225 269] [434 2287]]	[[2226 268] [434 2287]]	[[2224 270] [436 2285]]
NearestCentroid	[[2186 308] [1391 1330]]	[[2185 309] [1379 1342]]	[[2198 296] [1373 1348]]	[[2210 284] [1367 1354]]	[[2222 272] [1359 1362]]	[[2231 263] [1366 1355]]	[[2235 259] [1365 1356]]	[[2241 253] [1358 1363]]	[[2250 244] [1357 1364]]	[[2263 231] [1352 1369]]	[[2268 226] [1327 1394]]	[[2273 221] [1332 1389]]	[[2276 218] [1328 1393]]	[[2275 219] [1310 1411]]	[[2274 220] [1299 1422]]	[[2274 220] [1297 1424]]	[[2277 217] [1298 1423]]	[[2277 217] [1300 1421]]	[[2276 218] [1299 1422]]	[[2277 217] [1299 1422]]
NuSVC	[[1980 514] [534 2187]]	[[1982 512] [505 2216]]	[[2015 479] [507 2214]]	[[2038 456] [490 2231]]	[[2051 443] [474 2247]]	[[2059 435] [460 2261]]	[[2068 426] [450 2271]]	[[2073 421] [456 2265]]	[[2092 402] [462 2259]]	[[2103 391] [454 2267]]	[[2120 374] [439 2282]]	[[2126 368] [444 2277]]	[[2127 367] [443 2278]]	[[2137 357] [443 2278]]	[[2139 355] [437 2284]]	[[2139 355] [431 2290]]	[[2142 352] [433 2288]]	[[2146 348] [435 2286]]	[[2149 345] [441 2280]]	[[2149 345] [446 2275]]
PassiveAggressiveClassifier	[[1770 724] [777 1944]]	[[2066 428] [909 1812]]	[[2084 410] [875 1846]]	[[1629 865] [813 1908]]	[[1695 799] [814 1907]]	[[1580 914] [742 1979]]	[[1810 684] [738 1983]]	[[1540 954] [713 2008]]	[[2205 289] [807 1914]]	[[2211 283] [780 1941]]	[[1987 507] [646 2075]]	[[2179 315] [671 2050]]	[[1961 533] [578 2143]]	[[1631 863] [544 2177]]	[[1615 879] [490 2231]]	[[1681 813] [503 2218]]	[[1676 818] [503 2218]]	[[1680 814] [501 2220]]	[[1683 811] [491 2230]]	[[1692 802] [489 2232]]
Perceptron	[[1979 515] [829 1892]]	[[2023 471] [849 1872]]	[[1984 510] [826 1895]]	[[2038 456] [1080 1641]]	[[2127 367] [839 1882]]	[[2096 398] [756 1965]]	[[2070 424] [729 1992]]	[[1894 600] [728 1993]]	[[2061 433] [710 2011]]	[[2117 377] [664 2057]]	[[2043 451] [691 2030]]	[[2006 488] [657 2064]]	[[1652 842] [563 2158]]	[[2046 448] [592 2129]]	[[2121 373] [665 2056]]	[[1948 546] [389 2332]]	[[2203 291] [764 1957]]	[[2133 361] [611 2110]]	[[2086 408] [659 2062]]	[[1893 601] [441 2280]]
QuadraticDiscriminantAnalysis	[[2142 352] [1419 1302]]	[[2172 322] [1512 1209]]	[[2130 364] [1420 1301]]	[[2198 296] [1438 1283]]	[[2218 276] [1478 1243]]	[[2203 291] [1395 1326]]	[[2231 263] [1435 1286]]	[[2137 357] [1336 1385]]	[[2244 250] [1424 1297]]	[[2058 436] [1152 1569]]	[[2265 229] [1402 1319]]	[[2256 238] [1350 1371]]	[[2287 207] [1480 1241]]	[[2277 217] [1398 1323]]	[[2253 241] [1301 1420]]	[[2279 215] [1422 1299]]	[[2281 213] [1328 1393]]	[[2282 212] [1338 1383]]	[[2281 213] [1324 1397]]	[[2027 467] [1074 1647]]
RandomForestClassifier	[[1826 668] [222 2499]]	[[1839 655] [231 2490]]	[[1887 607] [176 2545]]	[[1916 578] [165 2556]]	[[1928 566] [145 2576]]	[[1949 545] [121 2600]]	[[1962 532] [110 2611]]	[[1994 500] [105 2616]]	[[2006 488] [101 2620]]	[[2023 471] [87 2634]]	[[2046 448] [89 2632]]	[[2069 425] [86 2635]]	[[2094 400] [79 2642]]	[[2100 394] [75 2646]]	[[2111 383] [52 2669]]	[[2112 382] [52 2669]]	[[2113 381] [51 2670]]	[[2122 372] [52 2669]]	[[2112 382] [54 2667]]	[[2115 379] [55 2666]]
RidgeClassifier	[[1904 590] [800 1921]]	[[1921 573] [780 1941]]	[[1941 553] [779 1942]]	[[1951 543] [771 1950]]	[[1954 540] [764 1957]]	[[1954 540] [757 1964]]	[[1960 534] [751 1970]]	[[1970 524] [754 1967]]	[[1990 504] [740 1981]]	[[1997 497] [725 1996]]	[[2021 473] [711 2010]]	[[2018 476] [712 2009]]	[[2022 472] [704 2017]]	[[2031 463] [675 2046]]	[[2055 439] [639 2082]]	[[2056 438] [635 2086]]	[[2051 443] [627 2094]]	[[2056 438] [622 2099]]	[[2056 438] [619 2102]]	[[2058 436] [625 2096]]
RidgeClassifierCV	[[1904 590] [800 1921]]	[[1921 573] [780 1941]]	[[1941 553] [779 1942]]	[[1952 542] [771 1950]]	[[1955 539] [763 1958]]	[[1954 540] [757 1964]]	[[1960 534] [751 1970]]	[[1970 524] [754 1967]]	[[1990 504] [740 1981]]	[[1997 497] [725 1996]]	[[2020 474] [711 2010]]	[[2018 476] [713 2008]]	[[2022 472] [704 2017]]	[[2031 463] [675 2046]]	[[2055 439] [639 2082]]	[[2056 438] [635 2086]]	[[2053 441] [627 2094]]	[[2056 438] [622 2099]]	[[2057 437] [619 2102]]	[[2058 436] [625 2096]]
SGDClassifier	[[2097 397] [877 1844]]	[[2126 368] [836 1885]]	[[2089 405] [708 2013]]	[[2137 357] [749 1972]]	[[2121 373] [702 2019]]	[[2134 360] [703 2018]]	[[2103 391] [620 2101]]	[[1974 520] [442 2279]]	[[2168 326] [673 2048]]	[[2214 280] [684 2037]]	[[2194 300] [612 2109]]	[[2234 260] [607 2114]]	[[2229 265] [549 2172]]	[[2256 238] [573 2148]]	[[2133 361] [397 2324]]	[[2228 266] [445 2276]]	[[2252 242] [504 2217]]	[[2243 251] [471 2250]]	[[2215 279] [376 2345]]	[[2230 264] [487 2234]]
SVC	[[1981 513] [501 2220]]	[[1982 512] [464 2257]]	[[2013 481] [433 2288]]	[[2036 458] [426 2295]]	[[2047 447] [381 2340]]	[[2062 432] [358 2363]]	[[2066 428] [340 2381]]	[[2073 421] [331 2390]]	[[2084 410] [317 2404]]	[[2101 393] [309 2412]]	[[2121 373] [280 2441]]	[[2147 347] [273 2448]]	[[2154 340] [264 2457]]	[[2163 331] [249 2472]]	[[2170 324] [235 2486]]	[[2170 324] [228 2493]]	[[2171 323] [228 2493]]	[[2168 326] [232 2489]]	[[2170 324] [230 2491]]	[[2171 323] [235 2486]]
XGBClassifier	[[1892 602] [301 2420]]	[[1932 562] [287 2434]]	[[1951 543] [267 2454]]	[[1974 520] [250 2471]]	[[1991 503] [218 2503]]	[[2010 484] [200 2521]]	[[2036 458] [187 2534]]	[[2028 466] [200 2521]]	[[2041 453] [168 2553]]	[[2061 433] [166 2555]]	[[2093 401] [133 2588]]	[[2107 387] [151 2570]]	[[2125 369] [132 2589]]	[[2153 341] [127 2594]]	[[2165 329] [107 2614]]	[[2164 330] [93 2628]]	[[2166 328] [106 2615]]	[[2169 325] [104 2617]]	[[2161 333] [104 2617]]	[[2165 329] [104 2617]]

Appendix B: Code Snippets

The Python code snippet 1 is used to create the dfstuDemo DF, a subset of which is shown in table 4.10, which contains the students' demographics, as well as their registration details.

```
assert(dfstuI['id_student'].equals(dfstuR['id_student']))
dfstuDemo = dfstuI.join(dfstuR['date_registration'])
dfstuDemo
```

Code Snippet 1: Merges dfstuI and dfstuR DataFrames.

The Python code snippet 2 is used to generate Figure 4.2, which shows a breakdown of the count of each type of activity in the dfvle DF.

```
dfvle.activity_type.value_counts()
```

Code Snippet 2: Counts the different VLE activities.

The Python code snippet 3 is used to generate Table 4.11, which shows the dfstuVSumD DF with the aggregate sum_click per day.

```
dfstuVSumD = (dfstuV.groupby(['id_student','code_module','code_presentation','date'])
              ['sum_click'].sum().reset_index())
dfstuVSumD
```

Code Snippet 3: Aggregates the sum_click per day.

The Python code snippet 4 is used to generate Figure 4.3, which shows the start and end dates in days, of the VLE interactions.

```
dfstuVSumD.date.min()
dfstuVSumD.date.max()
```

Code Snippet 4: Determines the module start and end dates.

The Python code snippet 5 is used to generate Table 4.12 with the dfstuVSumD DF, showing the new week attribute column.

```
dfstuVSumD.loc[:,'week']=(dfstuVSumD['date']//7)+1
dfstuVSumD.loc[dfstuVSumD['week'] < 0,'week']= 0
dfstuVSumD
```

Code Snippet 5: Create equivalent week numbers from day numbers.

The Python code snippet 6 is used to generate Figure 4.4, which shows the start and end weeks, of the VLE interactions.

```
dfstuVSumD.week.min()
dfstuVSumD.week.max()
```

Code Snippet 6: Determine the module start and end week.

The Python code snippet 7 is used to generate Table 4.13, which shows multi-index dfstuVSumW DF, with aggregated sum_click values for 40 weeks.

```
dfstuVSumW = (dfstuVSumD.pivot_table(index=['id_student','code_module','code_presentation'],
                                     columns="week", values="sum_click",aggfunc=sum,fill_value=0))
dfstuVSumW
```

Code Snippet 7: Aggregate sum_click values on a weekly basis.

The Python code snippet 8 is used to generate Table 4.14, which shows the flattened dfstuVSumW DF, with aggregated sum_click values for 40 weeks.

```
dfstuVSumW = pd.DataFrame(dfstuVSumW.to_records())
dfstuVSumW
```

Code Snippet 8 Flatten multi-index DataFrame to a single index DataFrame.

The Python code snippet 9 is used to generate Table 4.15, which shows a subset of dfFinalCumU DataFrame, with demographic and weekly VLE data.

```
dfFinalCumU = pd.merge(dfstuDemo,dfstuVSumW)
dfFinalCumU
```

Code Snippet 9: Merge dfstuDemo and dfstuVSumW DataFrames.

The Python code snippet 10 is used to generate Figure 4.7, which shows the missing values in the dfFinalCumU DF.

```
dfNulls[dfNulls>0]
```

Code Snippet 10: Identify variables with missing values.