University of BRISTOL

DEPARTMENT OF ENGINEERING MATHEMATICS

Comparative Analysis of Machine-Learning Models for Student Dropout Prediction in a Virtual Learning Environment

Incorporating Student Engagement and Socioeconomic Features

Carlos Duran Calle

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering.

Friday 29th August, 2025

Supervisor: Dr. Felipe Campelo

Declaration

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of MSc in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Carlos Duran Calle, Friday 29th August, 2025

Contents

List of Figures

List of Tables

List of Algorithms

‘

Abstract

In virtual learning environments, high attrition rates continue to be a problem. A comparative machine-learning pipeline was created to detect students who might withdraw early and facilitate prompt intervention techniques in order to solve this problem. The Open University Learning Analytics Dataset was used for the analysis, and the student outcomes were classified as *Withdrawn* (class 0), *Fail* (class 1), and *Pass* (class 2). Before the initial test, three easy-to-understand engagement indicators were created: *excellent score* (≥ 70), *active in VLE*, and a composite *student engagement* flags. To add contextual depth, socioeconomic and demographic factors (highest education, IMD band, disability, age, and region) were also included. To maintain class proportions across training and testing sets, data splitting used a cohort and outcome stratified approach. Random Forest, Multinomial Logistic Regression, K-Nearest Neighbours, LightGBM, Support Vector Machine, and Neural Network were the six machine learning models that were assessed. Custom weights (class 0: 2.09, class 1: 1.31, and class 2: 0.48) were used in conjunction with dropout-focused scoring metrics to address class imbalance. Stratified 5-fold GridSearchCV was used for hyperparameter optimisation. From data ingestion to model selection, the entire process adhered to a comprehensive pipeline. With an approximate 66.8% recall, efficient training time ($\sim$ 4 minutes), and interpretable coefficients, Multinomial Logistic Regression showed excellent performance in identifying *Withdrawn* students, qualifying it for early-warning applications. In order to reduce the possibility of missing students who might withdraw, the modelling approach gave recall a higher priority than precision.

A complete, modular codebase with extensive documentation was created. Core Python modules, encoded train-test splits, saved models and evaluation metrics, structured directories for raw and processed data, and visualisation outputs are all part of the project structure. Every notebook provides a thorough, model-specific analysis that makes it possible to replicate results and compare different strategies. For openness and cooperation, the entire repository is available to the public on GitHub.

Supporting Technologies

This section presents a detailed summary, in bullet point form, of any third-party resources used during the project:

- I looked up model implementation strategies and best practices on the *scikit-learn* documentation website at:
  `https://scikit-learn.org/stable/`
- I used a number of *Python* public-domain libraries for data science and machine learning, including:
  - *Pandas* and *NumPy* for numerical operations and data manipulation;
  - *Seaborn* and *Matplotlib* for statistical data visualisation; and
  - *scikit-learn* for machine learning implementation, including:
    - Random Forest, Logistic Regression, K-Nearest Neighbours, Support Vector Machine, and Multi-layer Perceptron (Neural Networks).
    - GridSearchCV for 5-fold cross-validation and hyperparameter optimisation.
    - Modules for preprocessing (data encoding and stratification), evaluation metrics, and custom scoring functions.
  - *LightGBM* for gradient boosting machine learning algorithms;
  - *SciPy* for statistical functions and analysis;
  - *joblib* for model persistence and storing trained models.
- For interactive data analysis and model development across several analysis notebooks, I utilised the *Jupyter Notebook* environment.
- I coded every part of the project using *Visual Studio Code* as my primary integrated development environment.
- For version control and project repository hosting, I utilised *GitHub*, which is accessible to the public at:
  `https://github.com/MScProjs/MScProject2025-Carlos-Duran-Calle`
- My main source of data was the *Open University Learning Analytics Dataset (OULAD)*, which I obtained from:
  `https://analyse.kmi.open.ac.uk/open-dataset`
- I utilised LaTeX to format my dissertation report using the desktop application *TeXstudio*.
- I created pipeline diagrams and visual representations of the methodology using *Microsoft PowerPoint*.

Notation and Acronyms

SDP	:	Student Dropout Predictor
MOOC	:	Massive Open Online Courses
VLE	:	Virtual Learning Environments
OULAD	:	Open University Learning Analytics Dataset
OU	:	Open University
ML	:	Machine Learning
RF	:	Random Forest
LR	:	Logistic Regression
LightGBM	:	Light Gradient Boosting Machine
NN	:	Neural Networks
KNN	:	K-Nearest Neighbours
SVM	:	Support Vector Machine
SE	:	Student Engagement
IMD	:	Index of Multiple Deprivation
TMA	:	Tutor Marked Assessment
TMA1	:	First TMA

Acknowledgements

I would like to start by expressing my heartfelt appreciation to my dissertation supervisor, Dr. Felipe Campelo, for his expert guidance and supportive approach throughout this dissertation. His valuable suggestions for enhancing my project were certainly significant in achieving the completion of this project.

I want to take this opportunity to express my sincere gratitude to my MSc Data Science coursemates and valued friends in Bristol for their constant backing during this journey. Your helpful suggestions and encouraging words have been precious to me. I appreciate your friendship and support. Wishing you the very best in all your future projects beyond this master's programme.

I would like to extend my sincere gratitude to PRONABEC, the Peruvian government institution that funded my studies in the UK. This unique funding enabled me to broaden my academic perspectives in a global context, for which I am extremely grateful.

Lastly, thank you to my family for believing in me and loving and supporting me no matter how far away you are. Your continual support has meant the world to me. I can't wait to meet you again and have some wonderful Peruvian food with you.

Chapter 1

Introduction

Education is now more widely accessible thanks to online learning. However, high attrition has continued. The median completion rate for massive open online courses (MOOCs) is close to 12.6%, which has frequently remained low [1]. Large-scale attrition has also been documented in open-university environments. A study from the Open University (UK) indicates that course-level dropout rates can reach as high as 78% [2]. These numbers have inspired timely assistance and predictive systems.

The Open University (OU) released the widely used Open University Learning Analytics Dataset (OULAD), a public dataset for learning analytics research. OULAD contains information from 22 module presentations and $32,593$ registered students. Demographics, tests, and daily click-log summaries from the Virtual Learning Environment (VLE) are among these data [3]. It is currently regarded as the de facto standard for feature engineering and early-warning models in distance learning because of its size and documentation [3]. For example, Hussain et al. [4] discovered that engagement signals predict outcomes consistently across platforms. The current study builds on this evidence by predicting engagement-aware early intervention in the OU context using OULAD.

One possible remedy for this issue is machine learning (ML). Large volumes of student data can be analysed using ML algorithms to find patterns that human observers might miss [5]. These algorithms can identify which students are most likely to leave the VLE early by analysing data on their click behaviour, assessment submission, and resource access [6]. Dropout prevention becomes predictive rather than reactive as a result of this predictive ability.

1.1 Problem Statement

The prediction task is structured as a three-class classification: *Pass*, *Fail*, and *Withdrawn*. This framing supports targeted responses (e.g., academic support for likely fail; re-engagement for likely withdrawn) and reflects OULAD's $final\_result$ taxonomy. This outcome schema and its connection to registration and assessment tables are described in detail in the OULAD documentation [3].

There is a noticeable disparity in class. The *Withdrawn* class made up 19.1% of the records in the project's processed dataset, making it the minority class for the three-way target. The other classes had 55.5% *Pass* and 25.4% *Fail*. In line with earlier findings that OULAD results are unequally distributed across *Pass/Fail/Withdrawn* categories, this imbalance makes model training and evaluation more difficult, particularly for recall on the minority class [7].

Early detection is also necessary. Delivering interventions before disengagement solidifies has a greater impact. The importance of early-phase risk detection in reducing withdrawal has been highlighted by previous work in open online courses [8]. As a result, the objective is to provide actionable, early warning in addition to final outcome prediction.

1.2 Research Objectives

According to earlier studies, student engagement is a significant predictor of both academic success and dropout, and engagement proxies that are derived from VLE activity traces have shown especially good results [4]. Similarly, when examined alongside engagement patterns, sociodemographic factors like age, region, disability, and prior education have been demonstrated to correlate with withdrawal [3].

The specific objectives of this project are:

- To incorporate the student engagement features as a meaningful predictor for the ML models.
- To determine which socio-demographic factors are significantly associated with student dropout.
- To choose a predictive model that can recognise students who are likely to drop out of a course early on.

The accomplishment is meaningful beyond the technical achievement. Accurate identification of dropout can also help institutions to determine the best investment of their support resources, develop more effective interventions and enhance the retention of students [9]. For students, that support can be the difference between meeting educational goals and falling among the dropout numbers.

1.3 ML Approach

This project uses and compares six widely used ML models with complementary advantages for dropout prediction: Random Forest (RF), an ensemble method known to be robust to high-dimensional data [10]; Logistic Regression (LR), to interpret the results for risk factor analysis [11]; K-Nearest Neighbours (KNN), to capture local data patterns based on instance-based learning [12]; LightGBM, optimized gradient boosting to handle large datasets efficiently [13]; Support Vector Machine (SVM), capable of dealing with high-dimensional spaces via kernel-based transformations [14]; and Neural Networks (NN), to model complex non-linear data relationships [15].

Every model will be tuned with respect to hyperparameters using GridSearchCV with 5-fold cross-validation. The main evaluation metric is recall of the *Withdrawn* class, as we attempt to identify all at risk students. This emphasis on minority class recall is paramount as missing out a student that requires assistance has higher stakes than sometimes offering help even if it may not be necessary [16].

1.4 Contributions

The present study builds upon existing research with three important advances. First, new engagement features are developed to model students' behaviour before their first attempt at the assessment. The *Student Engagement* (SE) variable is constructed with academic status along with VLE logs, similar to the approach of Hussain et al. [4]. It is demonstrated in this study that this composite measure allows better capturing student engagement than single traditional measures.

Second, socio-economic variables, in particular the Index of Multiple Deprivation (IMD), are added to explore to what extent dropout risk is influenced by exogenous factors. As compared with the other OULAD studies of Tomasevic et al. [17] and Hussain et al. [4] indicated demography, these relations were not further investigated, in contrast socio-economic aspects came out as some of the strongest drop-out predictors.

Third, it was developed a thorough model comparison framework with a unified evaluation schema. All models are trained on the same data split with data-specific class-weighting (*Withdrawn*: 2.09, *Fail*: 1.31, *Pass*: 0.48) to account for class imbalance, therefore fairly comparable and giving insights into a best approach for deployment.

1.5 Assumptions and Limitations

Assumptions In order to use the *excellent_score* flag and the composite *student_engagement* indicator while maintaining consistent ”pre-TMA1” time windows, this study assumes that the first Tutor-Marked Assessment (TMA1) results are released on or very near the official cut-off date. To ensure precise alignment of behavioural features and assessment periods, it also makes the assumption that the date offsets supplied in the OULAD dataset are trustworthy across all modules.

Limitations The results are based solely on the OULAD dataset, whose daily click-log granularity and UK-specific context may restrict external validity in other learning environments. With the exception of purely pre-assessment predictions, the models can only be used after *TMA1* grading because the *excellent_score* feature requires *TMA1* marks. Furthermore, the work was conducted over a three-month period on a personal laptop, which limited the scope of testing temporal windows beyond the *TMA1* stage, exploring alternative class weighting strategies, and conducting hyperparameter searches.

1.6 Challenges and Scope

Key challenges This study encounters three principal challenges: (i) class imbalance in the three outcome categories (*Pass, Fail, Withdrawn*), which could lead to bias against the minority *Withdrawn* class; (ii) the need for early prediction, since interventions work best when disengagement is found quickly; and (iii) balancing model accuracy with interpretability, so that stakeholders know why a student is flagged as *Withdrawn* student while still getting good predictions. Methodological risks encompass the prevention of temporal leakage among cohorts, the maintenance of fairness in the utilisation of socio-economic variables, and the evaluation of generalisability beyond a singular dataset.

Scope of this study The study employs the OULAD and conceptualises prediction as a three-class task. Some of the features are socio-demographic information, study load, and early engagement signals before the first test. A single pipeline is used to systematically compare six machine learning algorithms. Evaluation focusses on recall for the *Withdrawn* class, using weighted and macro metrics, as well as class weighting and custom scoring functions that focus on dropout.

Out of scope The project does not seek to establish causal inference, formulate interventions, conduct a fairness audit, or create time-to-event dropout models. External validation beyond OULAD and integration into institutional systems are reserved for subsequent initiatives.

Dissertation roadmap Chapter 2 gives the technical background, including educational data mining, OULAD, and metrics for classification that isn't balanced. Chapter 3 talks about the method, which includes preprocessing, feature engineering, stratification, weighting, custom metrics, and hyperparameter search. Chapter 4 shows the results of the experiments and compares the models. Chapter 5 talks about possible future research directions and extensions. Chapter 6 ends with contributions, limitations, and useful suggestions.

In summary, this project focuses on performing and comparing six ML models to identify the best-performing model for detecting students who have withdrawn from online courses. By emphasizing recall for the *Withdrawn* class and incorporating engagement and socio-economic factors, the study aims to provide actionable insights that support timely interventions and improve student retention in virtual learning environments.

Chapter 2

Technical Background

This chapter offers a theory based on student dropout prediction in VLE. Key concepts that comprise educational data mining, multi-class classification problems, and ML algorithms are discussed. The discussion is the technical basis for a comparison among dropout prediction models.

2.1 Educational Data Mining in VLE

Educational Data Mining in VLE explores the wealth of behavioural data produced by students (ie, clickstream patterns and resource access statistics) for the purpose of gaining knowledge on how they learn [18]. Browsing and resource navigation behaviour have been shown to be strong predictors of academic performance, such as course completion [19]. Combining this behavioural data with a learners' demographics and assessment results substantially enhances prediction accuracy of dropout models [20].

The temporal nature of this data is important. Early warning signs of at-risk students can be well-predicted from early-stage behaviour patterns [21]. The high-frequency of VLE data streams on the other hand requires advanced preprocessing to deal with missing values, irregular sampling intervals, and varying engagement patterns across different student populations [22]. In addition, integrating VLE interaction data with socio-economic features improves model stability and classification performance, and paves the way for a more systematic feature engineering in educational prediction systems [23].

2.2 The OULAD Dataset Architecture

OULAD is released as a set of CSV tables that can be joined through surrogate keys, enabling a student-centric relational view across demographics, registrations, assessments, learning materials, and VLE interactions [3], the database schema can be appreciated in the Figure 2.1 [24]. The public release contains 22 module-presentations, $32,593$ students, and $10,655,280$ daily click summaries, supporting at-scale analyses of behaviour and outcomes. All dates are stored as offsets relative to the module-presentation start, which simplifies time-window filters such as "before the first assessment".

2.2.1 Table Descriptions

Summaries below follow the official data descriptor for OULAD [3].

- Demographics and final result per student-module presentation (e.g., gender, age band, prior education, credits, disability, and $final\_result$).

- Registration and unregistration days for each student in each presentation; empty unregistration implies completion.

- Submission day and score (0–100) for each assessment attempted.

- Per-assessment metadata: type (TMA/CMA/Exam), cut-off day, and weight; non-exam assessments sum to 100 and exams are treated separately.

- Daily counts of interactions (*sum_click*) by student with specific learning materials (*id_site*).
- The catalog of VLE materials with activity type and planned availability windows (*week_from*, *week_to*).
- Module and presentation identifiers with presentation length in days.

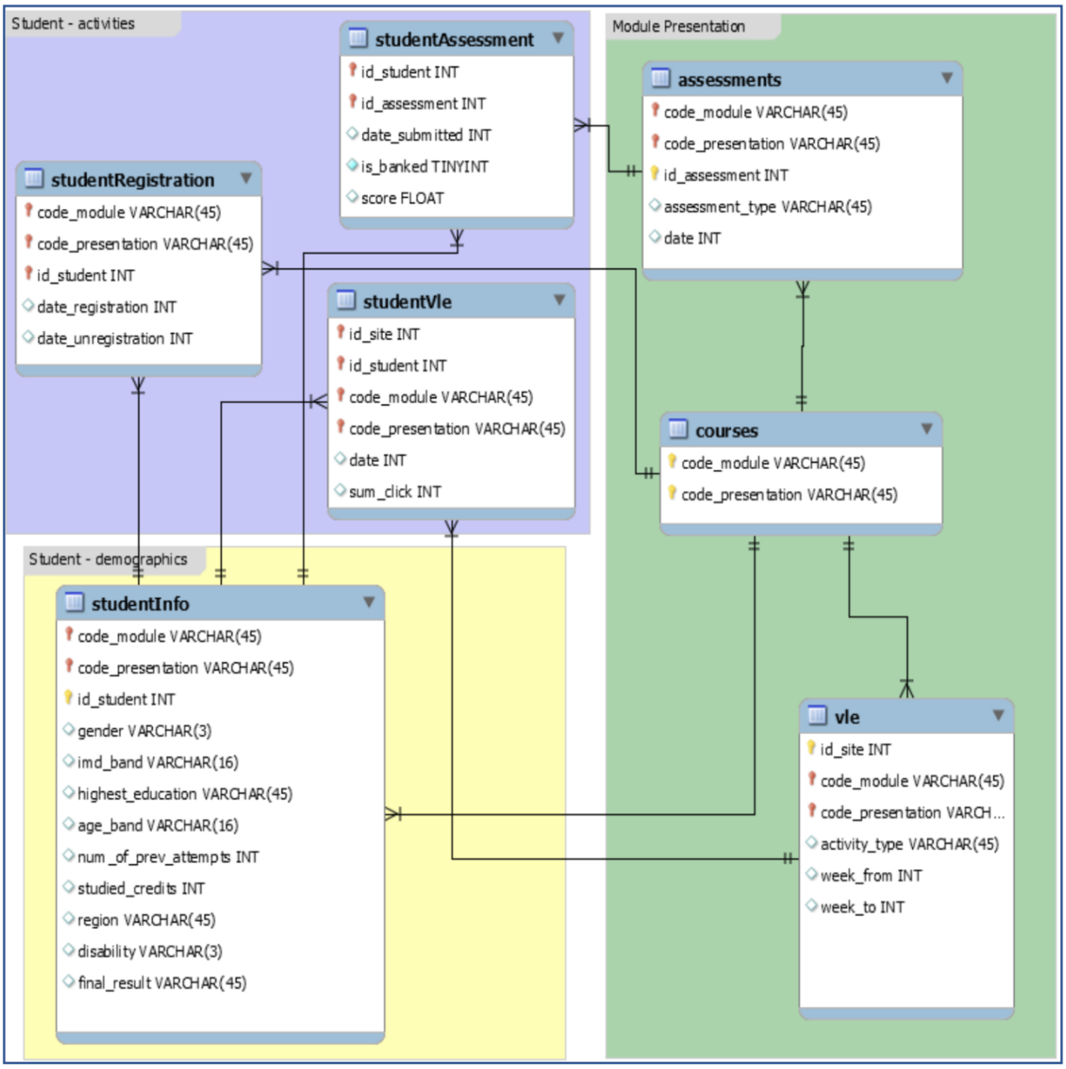

Figure 2.1: Database schema of OULAD dataset (reproduced from [24]).

2.3 Student Engagement Indicators

Simple, interpretable flags were engineered to capture early excellence and activity in ways that align with common early-warning practices in learning analytics [25]. Mathematical notations are added no understand following equations.

- Let $\mathbf{1}[\cdot]$ be the indicator function (1 if the condition is true, else 0).
- Let the first graded assessment in a presentation be indexed by a_1
- Let score a_1 be the student's first-assessment score (0–100).
- Let $date(a_1)$ be its cut-off day.
- Let C_{pre} be the student's total *sum_click* with $studentVle.date < date(a_1)$.

- Let μ_{pre} be the mean of C_{pre} within the same (*code_module*, *code_presentation*).
- These columns and date conventions are defined in OULAD.

2.3.1 Excellent Score Indicator

A binary indicator for early academic excellence was defined as:

$$\text{excellent\_Score} = \mathbf{1}[\,\text{score}(a_1) \geq 70\,] \tag{2.1}$$

According to Scholaro database, The Open University awards a “Merit” scale when is higher than 70% [26]. This threshold marks, where is categorised as distinction mark for OU, clearly high performance while keeping the rule easy to interpret; early assessment performance has been shown to be informative for final outcomes and targeted support.

2.3.2 Active in VLE Indicator

A binary indicator for above-average pre-assessment activity was defined as:

$$\text{active\_in\_VLE} = \mathbf{1}[\,C_{\text{pre}} > \mu_{\text{pre}}\,] \tag{2.2}$$

Clicks prior to the first assessment are counted and compared with the cohort’s mean for the same module-presentation; VLE click behaviours have repeatedly shown predictive value for course performance, including in OULAD-based studies [27].

2.3.3 Student Engagement Indicator

A composite engagement flag was defined with a logical OR:

$$\text{student\_engagement} = \mathbf{1}[\,\text{excellent\_Score} = 1 \;\vee\; \text{active\_in\_VLE} = 1\,] \tag{2.3}$$

This rule fires if either early excellence or above-average activity is observed, a common, conservative design for early-warning heuristics that favours recall of potentially successful or engaged students [25].

Why these signals The pair (grade-based, behaviour-based) captures complementary facets of engagement and aligns with evidence that clickstream engagement and formative performance jointly inform later achievement [28].

2.4 Stratification and Encoding

2.4.1 Cohort-Outcome Stratification

A stratified hold-out split was performed per cohort so each module–presentation kept its original class mix, and the target values (*Withdrawn/Fail/Pass*) was included in the strata to preserve outcome prevalence within every cohort. Class-imbalance concerns motivated stratification to reduce variance and avoid misleading metrics on under-represented outcomes [29]. The split was executed before any encoding to prevent information leakage from validation/test data back into training data [30]. The implementation relied on scikit-learn splitters with a fixed random state for reproducibility [31]. In the project, three alternatives were examined and Strategy 2 (cohort + outcome) was retained due to its better distribution in comparison with the Strategy 1 (cohort); a triple-key variant Strategy 3 which include “courses per term” was discarded due to tiny cells that break stratified sampling.

2.4.2 Feature Encoding

Encoding was intentionally minimal and applied only to a selected set of categorical variables that were found relevant for the SDP: *region*, *highest_education*, *imd_band*, *age_band*, and *disability*; numerical and binary features were left unchanged to keep the signal simple. Nominal fields were one-hot encoded to avoid imposing order, while ordered bands (*imd_band*, *age_band*) were ordinal-encoded to preserve ranks in a compact form [32]. Encoders were fit only on the training split and then applied to the test dataset, with unseen categories mapped safely to an “Unknown” bucket to avoid runtime errors and leakage [32]. The steps were orchestrated with *scikit-learn* library and the encoded matrices for the train and test dataset, and labels were persisted for downstream modelling.

2.5 Multi-Class Classification

In education, multi-class classification is complicated since the standard binary techniques are not effective in modelling the interactions among various performance levels, which leads to specialized algorithms [33]. Model collection is in turn influencing the ordinal nature of effects (e.g., from *Pass* to *Fail*) might need attentive attention at some stage in model training [34]. Standard decomposition methods may bias such type of ordered data, which validate why it is useful to apply native multi-class algorithms in order to predict student performances accurately [35].

Class imbalance is a critical challenge in educational data, as traditional accuracy metrics can be misleading by failing to reflect poor performance on key minority classes, such as *Withdrawn* students [36]. Proper evaluation of these models requires specialized metrics, such as class-specific recall and macro-averaged F1 scores, to ensure a balanced assessment [37]. Advanced techniques, including strategic oversampling combined with ensemble methods, have been shown to significantly improve the identification of these at-risk students [38].

2.6 Multi-Class Evaluation Metrics

The class-specific performance metrics can offer valuable insights on how a model performs and work well for imbalanced data. Among these, precision and recall are core indicators for assessment of classification effectiveness. For a given class i, precision is defined as the ratio of true positive predictions to all positive predictions, as shown in Equation 2.1. This concept is introduced in [16].

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i} \tag{2.4}$$

where TP_i represents true positives and FP_i represents false positives for class i.

On the other hand, recall is defined as the ratio of the actual positive instances that are classified as positive which is given in Equation 2.2 [16]. In educational applications, recall for the *Withdrawn* class is especially important as it measures the model's capacity to detect students who need early support, while having high recall rates ensures that such at-risk students are not missed out even with elevated false positive rates [39].

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i} \tag{2.5}$$

where TP_i represents true positives and FN_i denotes false negatives for class i.

Weighted metrics compensate for the class imbalance by incorporating the relative frequency of each class contributing to the overall performance measure and deliver a model effectiveness evaluation that is more representative of how well it generalizes to all categories [40]. Weighted recall is calculated according to the Equation 2.3 [41].

$$\text{Weighted Recall} = \sum_i w_i \times \text{Recall}_i \tag{2.6}$$

where w_i is the weight for class i, usually based on its frequency in the dataset.

Weighted F1-score is the harmonic mean of precision and recall for each class i [41] [42] as in the Equation 2.4. These balanced metrics allow us to measure the performance level for all categories of evaluation indicators while still being sensitive to the practical important of minority class detection in the educational intervention systems [35].

$$\text{Weighted F1} = \sum_i w_i \times F1_i \tag{2.7}$$

where $F1_i = 2 \times \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$ and w_i represents the weight assigned to class i.

2.7 Strategies for Imbalanced Data

Balanced class weight is one of the basic approaches to address the imbalance of the dataset where the importance of each class is adjusted automatically and inversely proportional to the frequency of that class and the mathematical foundation is given as it is written in the Equation 2.5 [43].

$$w_i = \frac{n_{\text{samples}}}{n_{\text{classes}} \times n_{\text{samples}_i}} \tag{2.8}$$

where w_i denotes the weight for class i, n_{samples} represents the total number of samples, n_{classes} indicates the number of classes, and n_{samples_i} represents the number of samples in class i.

Customized scoring metrics deepen this approach by considering domain-specific preferences, by dropout-specific optimization forcing the maximization of the recall for withdrawal detection via biasing the learning criterion in favour of at-risk students and focusing less on false negatives [44].

Early warning signs do exist to prevent high-risk youngsters and to intervene early. In the context of dropout prediction, the imbalance class is the minority dropout class, which results in heavy emphasize on recall to detect the dropout students for timely intervention [45]. Unbalanced-learning techniques (e.g., resampling or class weighting) generally improve recall for the rare class (dropout) at the expense of precision [44], which is the same precision-recall trade-off this work aims to find the best level of.

In the area of dropout early-detection, balancing the class distribution among minority and majority instances should be considered desirably by using class weights, which disadvantage the majority and advantage the minority class, e.g., the weight of the minority class being higher than 1 and the weight of the majority class being equal to 1 is common [46]. More recent losses, e.g., the Class-Balanced Loss, have tried to provide a more principled re-weighting beyond naïve inverses only when classes are long-tailed [47]. For the current study it is beneficial as we would like to investigate alternative to ad-hoc multipliers, used for balanced weighting, like x1.2 for the minority class *Withdrawn* and x0.8 for the majority class *Pass*. The multipliers are domain specific choices of the experimentations: there is one setting that uses no multipliers (balanced weighting), and one that applies the multipliers. Lastly, note that overall accuracy may be deceptive of imbalanced data, as Precision and Recall can better assess minority-class detection quality [48].

2.8 ML Algorithms for SDP

Ensemble learning exploits multiple models to reduce variance and improve generalization; in RF, bootstrap aggregation (bagging) draws resamples of the training set and averages randomized decision trees, providing robustness and out-of-bag error estimation [49] [10]. Feature importance in RF is commonly derived from impurity decreases or permutation-based accuracy drops, enabling screening of influential predictors in heterogeneous educational data [10]. Gradient-boosting frameworks fit weak learners sequentially under an additive model to minimize a differentiable loss, while LightGBM accelerates this process using histogram-based splits, leaf-wise growth, and gradient-based one-side sampling for scalability on high-dimensional, large cohorts [50] [13]. LR models the log-odds of class membership and estimates coefficients by maximum likelihood, yielding interpretable effects as odds ratios and supporting regularization for stability when features are correlated [51] [52]. SVM maximise the geometric margin via convex quadratic programming and use kernel mappings to induce flexible non-linear decision boundaries while controlling capacity through the margin and kernel parameters [14].

Instance-based learning with KNN classifies by a majority (plurality) vote among the K closest samples, so decision boundaries are local and sensitive to the choice of distance metric and K, with common metrics including Euclidean, Manhattan, and Mahalanobis distances [12] [53]. Multi-layer perceptrons (a type of NN) compose affine transformations with non-linear activations and are trained end-to-end by backpropagation to minimize a supervised loss, with universal approximation guaranteeing expressive capacity under sufficient width or depth [54] [55]. These foundations motivate comparing RF, LightGBM, LR, SVM, KNN, and NN for dropout prediction, balancing interpretability, non-linear modelling power, and computational efficiency under class imbalance typical of at-risk cohorts [35].

2.9 Model Evaluation and Comparison Methodologies

Performance was estimated with stratified 5-fold cross-validation [56] on the training split to preserve class proportions and reduce variance in imbalanced data, with a fixed stratified train–test split reused for all models to ensure comparability [37]. Hyperparameter search was executed within cross-validation (GridSearchCV) and final metrics were computed once on the untouched test split to limit selection bias [57]. All models were evaluated on identical folds and scoring rules to enable fair, paired comparisons across the pipeline [58]. Hyperparameters were optimized via exhaustive GridSearchCV with five folds, parallel execution, dropout-recall as the objective, and class-weighting to emphasize minority cases [56] [35].

Regularisation and early stopping are treated as crucial mechanisms to curb over-fitting and stabilise learning dynamics [59]. In LR, the strength of regularisation is controlled by the inverse-penalty parameter

C, with $L1/L2$ penalties shaping sparsity and shrinkage behaviour [60]. For SVM, the soft-margin constant C and kernel scale gamma are tuned to balance margin violations and function smoothness [14]. In decision trees and ensemble methods, maximum depth and the number of estimators are adjusted to limit variance and improve robustness [10]. In gradient boosting, the learning rate and number of boosting rounds jointly govern additive model capacity and error reduction [50]. For KNN, the neighbourhood size k and the choice of distance metric determine locality and boundary smoothness [12]. In multilayer perceptrons, hidden-layer size and weight decay are set under backpropagation to control capacity and enhance generalisation [61].

Chapter 3

Methodology and Implementation

This chapter describes the methodical process used to create and assess ML models for student dropout prediction, covering everything from data preprocessing to model deployment and validation.

3.1 Overview and Research Design

With an emphasis on early intervention capabilities, this study applies a ML pipeline intended to forecast the likelihood of student dropout in online learning environments. By creating predictive models that can identify at-risk students before they drop out of their courses, the study tackles the crucial problem of student retention in higher education.

3.2 Data Class Distribution

The class distribution of the cleaned and merged dataset, which comprises three outcome categories, is shown in Table 3.1. There are $27,725$ records in all in the dataset. There is a glaring class imbalance in the distribution: the frequency of *Fail* outcomes is closer to *Withdrawn* than to *Pass*, while the number of *Withdrawn* outcomes is much lower than that of *Pass* outcomes.

	Withdrawn (Class 0)	**Fail (Class 1)**	**Pass (Class 2)**
Counts	5,296	7,044	15,385
Percentages	19.1%	25.4%	55.5%

Table 3.1: Data Class Distribution

3.2.1 Research Methodology Framework

The approach uses supervised learning and compares six different ML in a methodical manner: RF, Multinomial LR, KNN, LightGBM Gradient Boosting, SVM, and NN. Robust evaluation across various learning paradigms, ranging from distance-based and deep learning approaches to ensemble methods and linear models, is ensured by this thorough algorithmic comparison.

From data intake to model winning selection, the research framework uses a structured pipeline (Figure 3.1), integrating modular Python architecture for scalability and reproducibility. Each algorithm undergoes rigorous hyperparameter optimisation using GridSearchCV with 5-fold cross-validation to guarantee fair comparison and optimal performance extraction.

3.2.2 Dropout Optimization

To enable successful early intervention systems, the main goal is to maximise dropout recall to $\geq 60\%$. Since false negatives, or missing *Withdrawn* students, have more serious repercussions than false positives, or failing students who might succeed, in educational intervention contexts, this metric gives priority to identifying students who will drop out. In order to balance practical resource constraints with effective

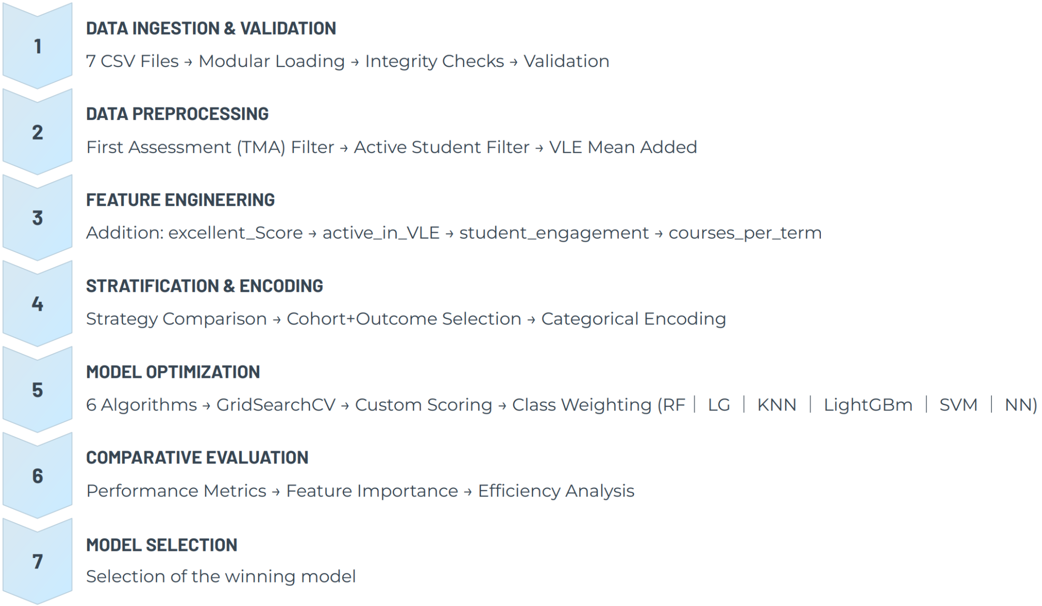
1
DATA INGESTION & VALIDATION
7 CSV Files → Modular Loading → Integrity Checks → Validation
2
DATA PREPROCESSING
First Assessment (TMA) Filter → Active Student Filter → VLE Mean Added
3
FEATURE ENGINEERING
Addition: excellent_Score → active_in_VLE → student_engagement → courses_per_term
4
STRATIFICATION & ENCODING
Strategy Comparison → Cohort+Outcome Selection → Categorical Encoding
5
MODEL OPTIMIZATION
6 Algorithms → GridSearchCV → Custom Scoring → Class Weighting (RF | LG | KNN | LightGBm | SVM | NN)
6
COMPARATIVE EVALUATION
Performance Metrics → Feature Importance → Efficiency Analysis
7
MODEL SELECTION
Selection of the winning model

Figure 3.1: The Seven-Phase Project Pipeline

early intervention capabilities, the 60% threshold was set especially for this project. This ensures that roughly two-thirds of *Withdrawn* students are identified while maintaining manageable caseloads for institutional support services.

3.2.3 Class Imbalance Challenge

With 19.1% dropout students (minority class), 25.4% failing students, and 55.5% passing students (majority class), the dataset represents a serious class imbalance issue. Specific optimisation techniques, such as unique class weighting schemes and dropout-focused evaluation metrics, are needed to address this imbalance. By using class weights of roughly 2.09x for dropouts, 1.31x for failures, and 0.48x for passes across various algorithms, the research employs customised strategies to make sure minority class detection is not overpowered by majority class dominance.

3.3 Data Architecture and Modular Design

As shown in Figure 3.1, the implementation uses a methodical seven-phase pipeline architecture intended for comprehensive SDP. From the intake of raw data to the deployment of models, this modular design guarantees repeatable processes while preserving data integrity across the ML pipeline.

3.3.1 Phase 1: Data Ingestion and Validation

Seven CSV files containing learning analytics data are automatically loaded at the start of the pipeline. Validation protocols check column structures and data integrity.

3.3.2 Phase 2: Data Preprocessing

Predictive modelling is based on core data transformations. By determining the earliest *TMA* (Tutor Marked Assessment) dates for each course-presentation combination, assessment filtering establishes standardised prediction windows for early intervention. Consistency across diverse data sources is ensured by data type conversions and missing value handling. Only participants who are actively participating during assessment periods are kept by student filtering protocols, which eliminate inactive participants.

3.3.3 Phase 3: Feature Engineering

From raw learning analytics, sophisticated feature creation algorithms produce predictive indicators. Before assessment deadlines, click-stream data is compiled by VLE engagement metrics to identify patterns in behaviour. Academic performance indicators establish engagement thresholds and binary flags for excellence (scores ≥ 70). The *courses_per_term* feature provides context for student capacity analysis by quantifying the distribution of academic load.

3.3.4 Phase 4: Stratification and Encoding

Advanced stratification techniques preserve demographic representation while maintaining class proportions across cohort boundaries. With 19.1% dropout representation, the chosen cohort+outcome strategy guarantees balanced train-test splits. The specific transformations used in categorical encoding are binary encoding for dichotomous variables, ordinal encoding for hierarchical relationships, and one-hot encoding for nominal variables.

3.3.5 Phase 5: Model Optimization

Using thorough hyperparameter optimisation, a systematic algorithm comparison assesses six different ML techniques. 5-fold cross-validation in GridSearchCV guarantees equitable comparison across various learning paradigms. Custom scoring metrics use specific weighting strategies to manage class imbalance while giving priority to dropout recall optimisation.

3.3.6 Phase 6: Comparative Evaluation

A thorough model evaluation analyses performance across six algorithms using a variety of visualisation techniques. Direct performance comparison is made possible by horizontal bar charts that compare key metrics (dropout recall, dropout precision, at-risk recall, weighted F1). Multi-metric performance profiles for each model are displayed using five-dimensional radar plots, which also highlight the models' advantages and disadvantages. Runtime efficiency scatter plots determine the best algorithms for various deployment scenarios by analysing trade-offs between computational cost and performance. With comprehensive confusion matrices for every algorithm, three-panel performance dashboards categorise models into three performance tiers: Excellent $\geq 60\%$, Good $40-60\%$, and Needs Work $< 40\%$.

3.3.7 Phase 7: Model Selection

The comparative analysis in Phase 6 is the source of the systematic evaluation criteria used in evidence-based model selection. Dropout recall performance ($\geq$ 60% threshold), computational efficiency, and deployment feasibility are given top priority during the selection process. With a 66.84% dropout recall, superior efficiency (4-minute training time), and interpretability advantages, LR is the best option. Runtime efficiency analysis, practical deployment considerations, and performance tier classification (Excellent, Good, Needs Work) are among the selection criteria. While underperforming algorithms (NN at 18.67% recall) are not taken into consideration for production, runner-up models (SVM with 63.09% recall) offer backup options. The final choice strikes a balance between operational needs for actual educational intervention systems and predictive performance.

3.4 Data Preprocessing and Feature Engineering

A systematic six-stage pipeline is implemented by the data preprocessing and feature engineering phases (Phase 2 and Phase 3 in Figure 3.1), which convert raw learning analytics into predictive features for early intervention systems (Figure 3.2). Prior to determining academic outcomes, this creates behavioural indicators and temporal prediction windows that capture patterns of student engagement.

3.4.1 Stage 1: Integrating Data Sources

The pipeline starts with the thorough integration of five main data sources: student registrations, VLE interaction logs, assessment structure and student assessment metadata, and demographic data. The original dataset includes over 10 million VLE interactions and $32,593$ student registrations across 22 courses.

3.4.2 Stage 2: Defining the Initial Assessment Period

For every course-presentation combination, this stage identifies the first *TMA* dates. In order to determine the earliest assessment dates for 22 courses, this processes 106 *TMA* assessments. By eliminating participants who withdrew prior to their initial assessment opportunity, active student filtering guarantees that attention is directed towards students who might profit from intervention. Consequently, $27,725$ students in total match these filters.

3.4.3 Stage 3: Extracting Behavioural Features

Up until the initial evaluation, this stage shows how three fundamental behavioural dimensions (VLE interaction, academic performance, and course load), were extracted. In order to determine the total and average clicks per student in relation to a course-presentation baseline, VLE interaction analysis analyses $2,587,468$ clicks that took place prior to the first assessment deadlines. Academic performance extraction uses a merit threshold analysis (threshold 70) and records the initial assessment results. Students taking multiple concurrent courses are identified through course load quantification, which also provides context for their academic capacity. For this data, the maximum number of concurrent courses is 2; it was not found students with three or more courses at the same presentation time.

3.4.4 Stage 4: Creating Indicator

The fourth step filters the $27,725$ students and converts continuous behavioural measures into interpretable binary indicators. Merit-level performance (≥ 70 points) is attained by $16,687$ students (60.2% of the total) according to the academic excellence classification ($excellent\_Score$). $9,870$ students (35.6%) whose platform engagement surpasses course-specific averages are flagged by the VLE activity classification ($active\_in\_VLE$). Additionally, a course load analysis ($courses\_per\_term$) was added to quantify the academic workload of $1,270$ students (4.6%) who are managing multiple concurrent courses.

3.4.5 Stage 5: Aggregating an Engagement Indicator

In stage five, various pathways to academic success are captured by combining individual indicators into composite measures. By using logical OR operations to combine academic excellence and VLE activity, the unified engagement feature ($student\_engagement$) finds $19,932$ students (71.9%) who are either performing well or showing strong platform engagement. This composite approach acknowledges that various engagement patterns may lead to students' success.

3.4.6 Stage 6: Engineering the Final Training Dataset

The last step of data preparation creates a validated, ML-ready dataset of $27,725 students$ and 25 features by combining preprocessed data with engineered behavioural and demographic features. The distribution of classes in this dataset is 55.5% *Pass*, 25.4% *Fail*, and 19.1% *Withdrawn*. With the *Withdrawn* students representing a crucial minority class, it draws attention to the problem of class imbalance. Now that the dataset has been cleaned and optimised, it is ready for the next round of ML pipelines, where specific techniques will be used to predict outcomes for this important minority group.

3.5 Data Stratification Strategy

Three stratification techniques are evaluated methodically to maximise the preservation of class distribution. Using course-presentation combinations, Strategy 1 uses basic cohort-based stratification. Using a combination of cohort and outcome stratification, Strategy 2 maintains the distributions of academic outcomes and demographics. Triple stratification using cohort+outcome+courseload variables is attempted in Strategy 3.

Strategy Selection: Comparative analysis shows that Strategy 2 (Cohort + Outcome) is the best course of action. Since many cohort-outcome-courseload combinations contain only one student, triple stratification (Strategy 3) fails because combined groupings have insufficient sample sizes. As shown in Table 3.2, Strategy 2 effectively preserves the crucial 19.1% dropout representation across train-test splits while maintaining class proportions.

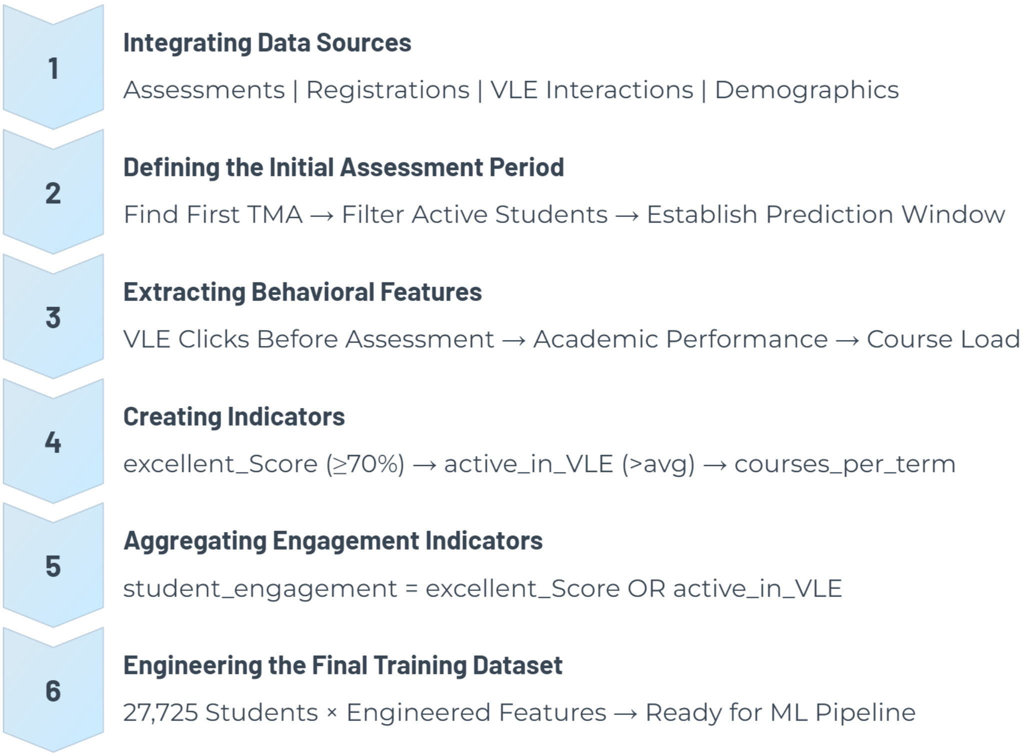
1
Integrating Data Sources
Assessments | Registrations | VLE Interactions | Demographics
2
Defining the Initial Assessment Period
Find First TMA → Filter Active Students → Establish Prediction Window
3
Extracting Behavioral Features
VLE Clicks Before Assessment → Academic Performance → Course Load
4
Creating Indicators
excellent_Score (≥70%) → active_in_VLE (>avg) → courses_per_term
5
Aggregating Engagement Indicators
student_engagement = excellent_Score OR active_in_VLE
6
Engineering the Final Training Dataset
27,725 Students × Engineered Features → Ready for ML Pipeline

Figure 3.2: The Six-Stage Engineering Feature Pipeline

Stratified Sampling Implementation: The chosen strategy guarantees equitable representation for both academic results and course cohorts. The train-test split allocation maintains proportional class representation by using a 80% – 20% distribution. This approach ensures sufficient samples for minority class optimisation in later model training stages while maintaining demographic diversity.

Strategy	**Withdrawn (Class 0)**	**Fail (Class 1)**	**Pass (Class 2)**	**Status**
Original Dataset	19.1%	25.4%	55.5%	Baseline
Strategy 1: Cohort Only	19.3%	25.0%	55.7%	Acceptable
Strategy 2: Cohort + Outcome	19.1%	25.4%	55.5%	**Selected**
Strategy 3: Cohort + Outcome + Courseload	-	-	-	Failed

Table 3.2: Class Distribution Comparison Across Stratification Strategies

3.6 Feature Selection for ML Models

In order to find variables with significant predictive relationships for predicting student outcomes, the feature selection process used exploratory data analysis. It also methodically looked at socio-demographic traits and engineered features to maximise model performance while preserving interpretability. Table 3.3 provides more information on these variables.

Six socio-demographic factors were subjected to linear regression analysis in order to measure their associations with student outcomes. Proportional distributions among the three outcome classes (*Pass, Fail, and Withdrawn*) were examined, and R^2 coefficients and slope measurements were used to evaluate the relevance of the variables. Table 3.4 displays a summary of the regression's results with R^2 and the corresponding slope for each class and variable selection.

Additional visual information about this regression can be found in Appendix A, which displays six plots for these socio-demographic variables. A summary of the findings from those plots is included here. The analysis showed that gender had perfect mathematical relationships (R^2 1.000) but minimal practical differences between female and male students (18.0% vs 20.0% withdrawal rates, 57.1% vs 54.1% pass

Variable	Type	Categories
Gender	Binary	Female, Male
Age Band	Ordinal	0–35, 35–55, 55+
Disability	Binary	No, Yes
Highest Education	Ordinal	No Formal → Post Graduate (5 levels)
IMD Band	Ordinal	10 income deprivation percentiles
Region	Nominal	13 UK geographical regions

Table 3.3: Socio-Demographic Variable Characteristics

rates), which led to its exclusion from model training. Strong correlations were found between Age Band (R^2 0.993 − 1.000), with older students exhibiting increasingly better results. Pass rates rose sharply from 53.5% to 68.2%, while withdrawal rates dropped from 19.5% (0 − 35 years) to 16.9% (55+ years). With pass rates rising from 38.7% to 70.9% and withdrawal rates falling from 25.6% (no qualifications) to 17.3% (post-graduate), the strongest substantive predictor (R^2 0.788 − 0.984) was highest education. Strong socioeconomic relationships were found by IMD Band (R^2 0.819 − 0.948), with pass rates rising from 43.6% to 65.0% as deprivation decreased and withdrawal rates for students from the most deprived areas being 22.0% compared to 16.3% in the least deprived areas. Disability Status demonstrated perfect mathematical relationships (R^2 1.000) with significant practical implications; students with disabilities had lower pass rates (45.8% vs. 56.5%) and much higher withdrawal rates (27.3% vs. 18.2%). Although region showed little variation and weak relationships (R^2 0.010−0.040), it was kept to account for regional differences in educational access that were not represented by other variables.

Variable	**Class**						**Selection Decision**
	Withdrawn		**Fail**		**Pass**		
	R^2	**Slope**	**R^2**	**Slope**	**R^2**	**Slope**	
Gender	1.000	+0.020	1.000	+0.011	1.000	-0.030	**Excluded**
Age Band	1.000	-0.013	0.994	-0.061	0.996	+0.074	**Included**
Disability	1.000	+0.091	1.000	+0.018	1.000	-0.107	**Included**
Highest Education	0.788	-0.021	0.984	-0.060	0.966	+0.081	**Included**
IMD Band	0.819	-0.001	0.909	-0.002	0.948	+0.002	**Included**
Region	0.010	+0.000	0.036	+0.002	0.040	-0.002	**Included**

Table 3.4: Model fit R^2 and slope by variable across outcome classes.

By identifying behavioural and performance patterns not found in raw demographic data, the six-stage feature engineering pipeline (Figure 3.2) produced four key variables that complemented socio-demographic characteristics: *student_engagement* (binary composite engagement that meets either excellence or VLE activity criteria), *active_in_VLE* (binary above-average virtual learning engagement prior to first assessment), *excellent_Score* (binary academic merit threshold indicator for ≥ 70% performance), and *courses_per_term* (continuous academic load distribution reflecting study intensity patterns).

3.7 Categorical Encoding Strategy

Using the *encoding_utils* module, tailored encoding techniques handle various categorical variable types. Region variables are treated as nominal categories with no intrinsic ordering through one-hot encoding. Ordinal encoding is used to preserve hierarchical relationships across socioeconomic and demographic dimensions while maintaining ranked sequences for the variables of education, age, and IMD band. For dichotomous classification, disability status uses binary encoding. Maintaining feature alignment and preventing data leakage are achieved by encoding consistency across train-test splits. Table 3.5 provides an overview of this.

Dataset Export Systematic model comparison is made possible by encoded datasets exporting as standardised CSV files (*X_train_encoded.csv*, *X_test_encoded.csv*, *y_train.csv*, *y_test.csv*). The encoding procedure preserves the temporal validity set in earlier preprocessing stages while generating consistent feature matrices appropriate for a range of algorithm requirements.

Variable Type	Variables	Encoding Method
Nominal	Region	One-Hot Encoding
Ordinal	Education, Age, IMD Band	Ordinal Encoding
Binary	Disability	Binary Encoding

Table 3.5: Encoding Strategy Matrix

3.8 Class Weighting Strategy

Class imbalance in educational datasets, where *Withdrawn* students make up only 19.1% of the dataset population, requires strategic weighting methods that are implemented through a two-stage calculation process that combines custom multipliers with sklearn's balanced weighting. First, balanced weights are calculated using sklearn's $compute\_class\_weight('balanced')$ function to produce baseline values proportional to inverse class frequencies. Next, strategic multiplier application is applied to improve dropout detection capabilities. This is demonstrated in Algorithm 3.1, where the multiplier values (1.2x, 1.0x, 0.8x) were chosen to reduce majority class dominance by 20% and increase dropout class sensitivity by 20%. These multiplier values were selected especially for this project and can be altered to improve the recall metric or other pertinent metrics, depending on the stakeholders' domain expertise. With this setup, the class balance is changed from the initial balanced weights $\{0 : 1.74, 1 : 1.31, 2 : 0.60\}$ to the optimised custom weights $\{0 : 2.09, 1 : 1.31, 2 : 0.48\}$.

```
Input: $y\_train$, $classes$
Output: Custom class weights
$balanced\_weights \leftarrow compute\_class\_weight(`balanced', classes, y\_train)$
$custom\_weights \leftarrow \{$
    $0 : balanced\_weights[0] \times 1.2$ // Dropout class boost
    $1 : balanced\_weights[1] \times 1.0$ // Fail class unchanged
    $2 : balanced\_weights[2] \times 0.8$ // Pass class reduction
$\}$
Resulting weights: $\{0 : 2.09, 1 : 1.31, 2 : 0.48\}$
```

Algorithm 3.1: Computation of custom class weights for ML training.

3.8.1 Class Weight implementation

Three different categories based on native support capabilities are revealed by class weight implementation across six chosen algorithms, as shown in Table 3.6. Through built-in *class_weight* parameters that automatically modify loss functions and sample importance during training, RF, LG, LightGBM, and SVM offer direct implementation. Because of its distance-based architecture, KNN has limited implementation capabilities and does not support native class weights. Custom weights can be declared, but they are not successfully integrated during prediction. With no alternative strategies like weighted sampling or custom loss functions implemented in the examined notebooks, NN ($MLPClassifier$) is still unsupported and expressly lacks class weight functionality.

Algorithm	Native Support	Implementation Method	Usage Status
Random Forest	✓	`class_weight` parameter	**Active**
Logistic Regression	✓	`class_weight` parameter	**Active**
LightGBM	✓	`class_weight` parameter	**Active**
Support Vector Machine	✓	`class_weight` parameter	**Active**
K-Nearest Neighbours	×	Use another type of weighting	**Not applied**
Neural Networks	×	Not supported by MLPClassifier	**Not applied**

Table 3.6: Class Weight Support by ML model

3.9 Custom Dropout-Focused Scoring Metrics

The three-class imbalanced nature of student outcome prediction necessitates specialised metrics that emphasise early identification of *Withdrawn* students. This led to the development of custom dropout-focused scoring functions to supplement a baseline evaluation framework of seven traditional *sklearn* metrics that provide balanced perspectives on classification performance across precision, recall, and F1-score dimensions. Standard ML evaluation metrics prioritise overall classification accuracy. Table 3.7 provides an appreciation of these standard metrics.

To separate performance measurement for the minority *Withdrawn* class (Class 0) and *Fail* student populations in student dropout prediction, three specialised scoring functions were created. Its addition is also visible in the Algorithm 3.2, where the ML model to be used for training is the input estimator.

Metric	Function	Scope	Purpose
`accuracy`	Overall classification rate	Global	General performance measure
`f1_weighted`	Weighted F1-score	Multi-class	Class-size adjusted F1
`f1_macro`	Unweighted F1-score	Multi-class	Equal class importance
`precision_weighted`	Weighted precision	Multi-class	Class-size adjusted precision
`recall_weighted`	Weighted recall	Multi-class	Class-size adjusted recall
`precision_macro`	Unweighted precision	Multi-class	Equal class precision
`recall_macro`	Unweighted recall	Multi-class	Equal class recall

Table 3.7: Standard Evaluation Metrics

```
Input: $estimator$, $X$, $y$
Output: Custom scoring functions for Withdrawn and Fail students
Function dropout_recall($estimator, X, y$):
    $y\_pred \leftarrow estimator.predict(X)$
    return $recall\_score(y, y\_pred, labels = [0], average = `macro')$      // class 0 recall
Function dropout_precision($estimator, X, y$):
    $y\_pred \leftarrow estimator.predict(X)$
    return $precision\_score(y, y\_pred, labels = [0], average = `macro')$   // class 0 precision
Function at_risk_recall($estimator, X, y$):
    $y\_pred \leftarrow estimator.predict(X)$
    return $recall\_score(y, y\_pred, labels = [0, 1], average = `macro')$   // classes 0, 1 recall
```

Algorithm 3.2: Definition of Custom Dropout and At-Risk Scoring Functions

Primary Optimisation Metric Since educational institutions gain more from identifying 60% or more of potential dropouts with false positives than from achieving high precision while missing the majority of at-risk students, the *dropout_recall* metric was chosen as the primary scoring function for hyperparameter optimisation across all six algorithms, giving priority to the identification of students likely to withdraw for proactive intervention over prediction accuracy. By ensuring that models are optimised for student retention rather than general classification accuracy, this custom scoring approach facilitates direct algorithm comparison on the most important prediction task while preserving thorough evaluation across conventional metrics.

3.10 ML Model Selection and Optimization

Six ML algorithms were compared methodically and rigorously optimised for student dropout prediction during the model selection phase.

3.10.1 Algorithm Selection and Optimization

The following six algorithms are complementary in their ability to handle the complexities of educational data mining: RF for robustness to feature interactions, Multinomial LR for interpretable probabilistic outputs, KNN for local behavioural patterns, LightGBM for advanced gradient boosting, SVM for non-linear boundary detection, and NN for deep learning capabilities. In order to address the crucial importance of accurately identifying at-risk students, each algorithm underwent systematic hyperparameter optimisation using GridSearchCV with 5-fold cross-validation, prioritising dropout recall performance ($\geq 60\%$) while maintaining balanced accuracy through custom class weights: *Withdrawn* class weighted 2.09x, *Fail* class 1.31x, and *Pass* class 0.48x. Each ML model's hyperparameter search space is displayed in Table 3.8. The hyperparameter search encompassed 2,032 total parameter combinations using parallel processing with 4 CPU cores.

Algorithm	**Key Parameters**	**Key Search Range**	**Search Space Size**
RF	n_estimators max_depth min_samples_split class_weight	100–500 10–25 2–10 balanced / custom	5,120 combinations
LR	C (regularization) penalty solver max_iter	0.01–100 L1 / L2 / ElasticNet SAGA / liblinear 1000–5000	450 combinations
KNN	n_neighbors metric weights algorithm	3–21 euclidean / manhattan / minkowski uniform / distance ball_tree / kd_tree / brute	480 combinations
LightGBM	n_estimators learning_rate max_depth subsample	100–500 0.01–0.2 3–8 0.7–1.0	3,888 combinations
SVM	C kernel gamma degree	0.01–100 linear / RBF / polynomial scale / auto / numeric 2–4 (poly only)	24 combinations
NN	hidden_layer_sizes alpha (L2 penalty) learning_rate_init solver	(50)–(150) + multilayer 0.0001–1.0 0.0001–0.1 adam / lbfgs	120 combinations

Table 3.8: Hyperparameter search spaces and search sizes for each algorithm.

Chapter 4

Result Analysis

This chapter presents a detail analysis comparing six machine learning models (RF, LR, KNN, LightGBM, SVM, and NN) for student dropout prediction using the OULAD dataset, with primary focus on optimising dropout recall performance to enable effective early intervention strategies.

4.1 Class Distribution Analysis

In order to address the inherent class imbalance in the OULAD dataset and guarantee representative class distributions, the dataset was systematically divided using stratified sampling with a 80:20 split ratio ($22,178$ training samples, $5,544$ test samples), while maintaining the original class proportions across both subsets. Figure 4.1 provides a visual representation of this distribution. Both the training and test partitions had a proportionate representation of each outcome class. The *Withdrawn* class (Class 0), which had $4,237$ training samples and 1,059 test samples, had only 19.1% of students, which is crucial for dropout prediction.

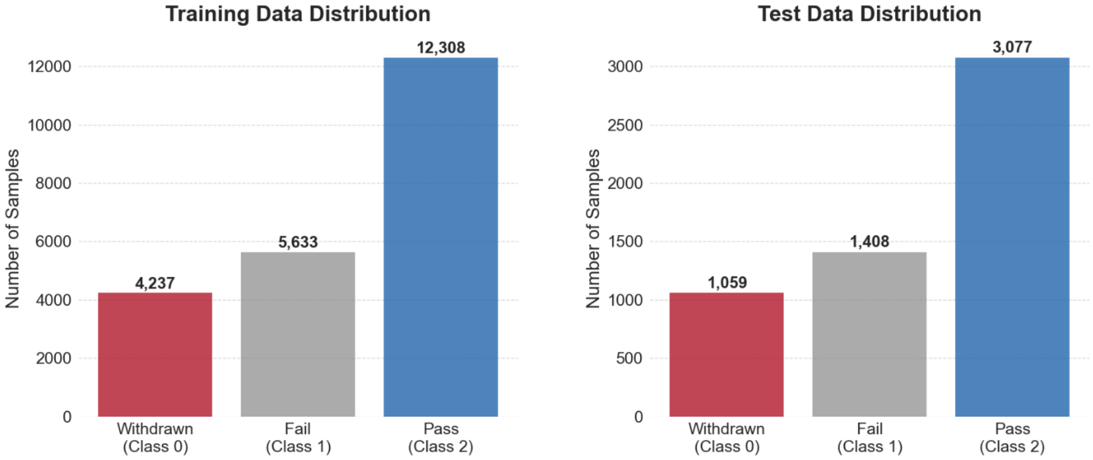

Figure 4.1: Data Class Distribution for Training and Test Data

4.2 Custom Weighted Analysis

There was a notable class imbalance and a strong bias in favour of successful students in the original training dataset. With 55.5% of the total samples, or $12,308$ students in the training set and $3,077$ in the test set, the *Pass* class (Class 2) dominated, as seen in Figure 4.1. With $5,633$ training samples and $1,408$ test samples, the *Fail* class (Class 1) made up 25.4% of the dataset. With $4,237$ training samples and $1,059$ test samples, the *Withdrawn* class (Class 0) represented only 19.1% of students, which is crucial for dropout prediction.

This distribution is further demonstrated in Figure 4.2, where the weighted distribution shows the optimisation strategy for dropout detection, and the pie chart emphasises the inherent imbalance. *Withdrawn* students (Class 0) received the highest weight of 2.09 according to the weighting method outlined in Section 3.8. Due to their at-risk status, *Fail* students (Class 1) were assigned a moderate weight of 1.31. With the lowest weight of 0.48, *Pass* students (Class 2) had less of an impact on model choices. Adjusted emphasis proportions of 53.9%, 33.8%, and 12.4% were the outcomes of these weights.

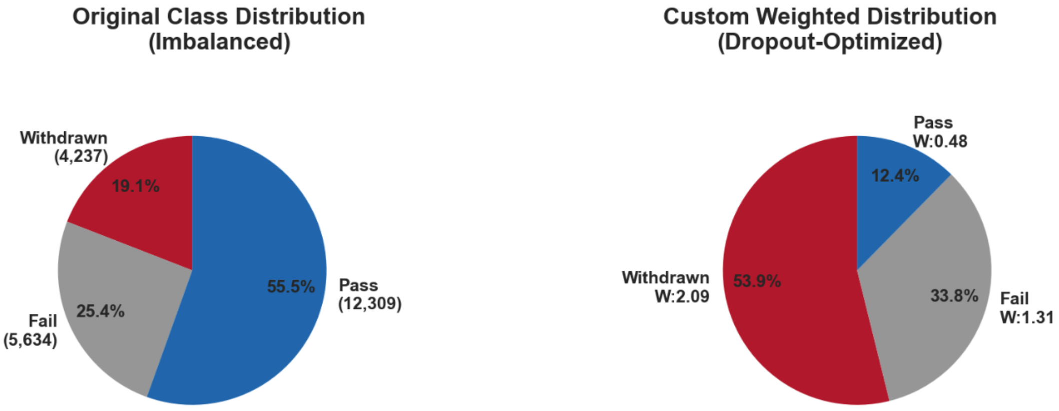

Figure 4.2: Original vs. Weighted Class Distribution

4.3 Model Performance Comparison and Rankings

4.3.1 Primary Metric Analysis: Dropout Recall Performance

Significant differences in dropout recall performance were found when six machine learning models were evaluated, creating distinct performance hierarchies for identifying student risk. When compared to the main optimisation metric of dropout recall, the models showed different performance tiers, according to 5-fold cross-validation analysis.

With a dropout recall score of 66.8%, LR was the best performer and had the strongest ability to identify dropout students, according to Figure 4.3. With this performance, the model showed strong predictive potential for early intervention systems, placing it well above the median threshold. SVM established a competitive alternative with comparable detection capacity, coming in second with 63.1% dropout recall. Three performance levels were identified by the analysis: (i) excellent ($\geq$ 60% dropout recall), attained by LR (66.8%) and SVM (63.1%); (ii) good (40–59%), represented by RF (59.0%) and LightGBM (56.0%); and (iii) needing improvement ($<$ 40%), with KNN (31.3%) and NN (18.7%), both of which did not support weight customisation during training. The significant performance differences between models—a 3.7-point difference between the top two and a 48.1-point difference between the best and worst performers—highlight the vital significance of selecting models carefully in student support systems.

4.3.2 Other Multi-Metrics Performance Evaluation

Proceeding with the examination of Figure 4.3, the assessment along several performance dimensions uncovered intricate trade-offs between classification accuracy and detection capability. Although dropout recall was the main optimisation goal, dropout precision, at-risk recall, and weighted F1 scores were also taken into account in order to provide a thorough evaluation of the overall efficacy of the model.

Although recall performance was poor, NN showed superior dropout precision (28.4%), indicating limited coverage of dropout students but high confidence in positive predictions. While LR recorded 25.0% precision, which represents the trade-off for maximised recall performance, RF (27.0%), LightGBM (26.9%), and SVM (26.2%) attained comparable precision levels.

In contrast to dropout recall, at-risk recall (which merges the *Withdrawn* and *Fail* classes into one) exhibited distinct rankings. The highest at-risk recall (40.8%) was attained by LightGBM, which was closely followed by SVM (39.2%), RF (39.3%), and LR (39.6%). This metric demonstrated the models'

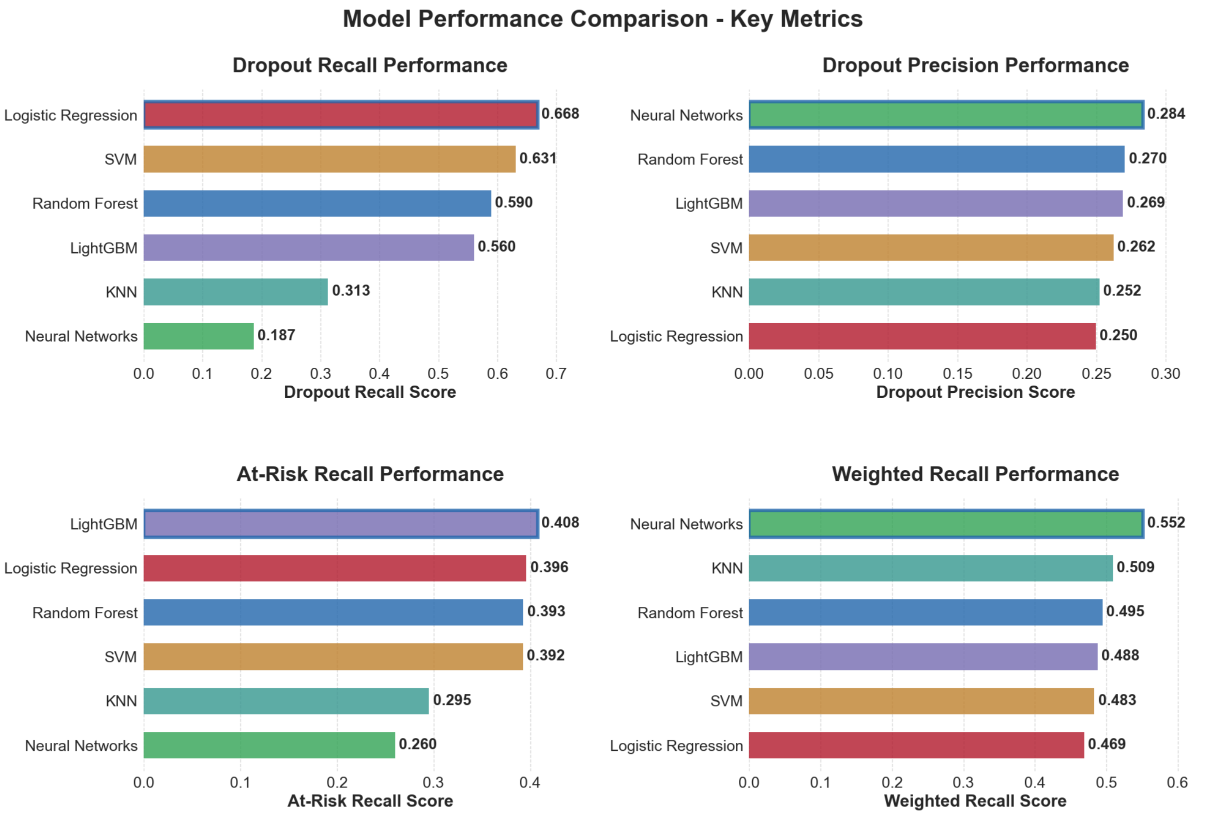

Figure 4.3: Model Performance Comparison

ability to detect students who need more academic assistance than just the immediate risk of dropping out.

Overall classification balance across all outcome categories was revealed by weighted F1 scores. The highest weighted F1 score (53.0%) was attained by NN, which was followed by KNN (50.7%) and LightGBM (50.5%). These findings showed that dropout recall-optimized models inevitably compromised overall classification balance, which is consistent with the intentional design decision to give dropout student identification priority.

No single model dominated across all performance dimensions, according to the multi-metric analysis. While models with strong overall classification performance showed limited dropout detection capabilities, the top-performing models for dropout recall (LR and SVM) performed moderately in other metrics.

4.4 Training Efficiency Analysis

Significant differences in the amount of time needed for training across machine learning algorithms were found by the computational efficiency analysis. Training durations varied from 4 minutes for LR to 7.14 hours for LightGBM, a difference of roughly 107 times in computational demands, as shown in Figure 4.4. LR achieved the highest dropout recall performance (66.8%) and showed remarkable efficiency with a 4-minute training duration. Because of this combination, the algorithm was positioned as the best option for quick development cycles and real-time model updates.

NN had the lowest dropout recall performance (18.7%) and required 50 minutes of training time. Out of all the algorithms that were evaluated, this efficiency-performance combination had the least favourable profile. SVM established a reasonable efficiency-performance balance for secondary deployment consideration, achieving competitive performance (63.1% dropout recall) with 1.34 hours of training time. Different computational requirements were shown by the intermediate performers. LightGBM required the longest training time of 7.14 hours for 56.0% performance, while RF used 3.13 hours with 59.0% dropout recall. The fact that KNN only achieved 31.3% dropout recall after 1.60 hours of training shows that there is no relationship between training time and performance results.

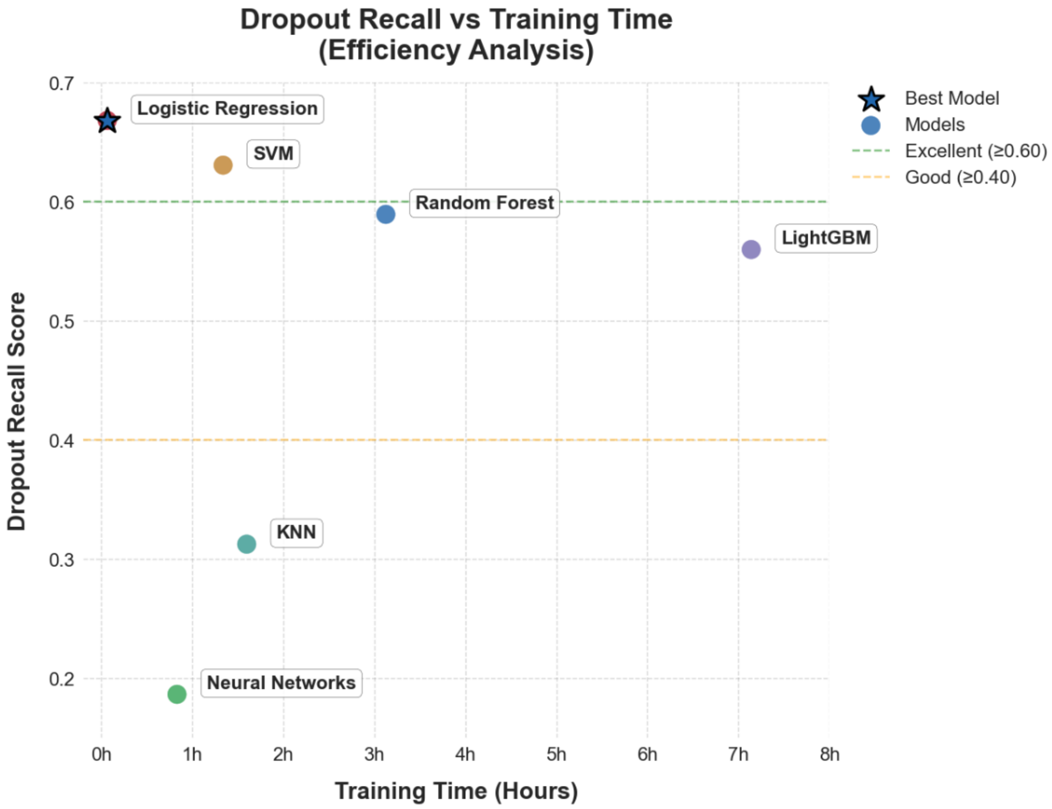

Figure 4.4: Dropout Recall vs. Training Time for Six ML Models

4.5 Hyperparameter Optimization Analysis

Different hyperparameter space dimensionalities and optimisation requirements were reflected in the wide variations in grid search complexity among algorithms. In order to evaluate the computational overhead related to model optimisation, the analysis looked at the number of parameter combinations tested during 5-fold cross-validation.

LightGBM demonstrated the highest optimisation complexity, requiring $3,888$ parameter combinations during grid search and resulting in the longest training duration of 7.14 hours, as illustrated in Figure 4.5. With $6,120$ combinations over 3.13 hours, RF came in second, demonstrating improved parameter evaluation efficiency in spite of a larger search space. Given the small but well-explored hyperparameter space, LR's practical advantage for quick deployment is highlighted by its optimal efficiency, testing only 450 parameter combinations in 4 minutes. Although the additional computational investment did not result in performance gains, NN evaluated 120 combinations in 50 minutes, indicating moderate optimisation complexity. Lastly, SVM demonstrated targeted hyperparameter optimisation that produced efficient performance results by testing 24 combinations in 1.34 hours.

An inverse relationship between search complexity and performance efficiency was found by the optimisation intensity analysis. LightGBM and RF, two algorithms that required a great deal of hyperparameter exploration, used disproportionate amounts of computational resources without offering any performance benefits over more straightforward methods.

4.6 Multi-Dimensional Performance

A five-dimensional radar plot analysis is presented in Figure 4.6, which offers a comprehensive visual depiction of model performance across all evaluation metrics at the same time. An alternate viewpoint for comparing the metrics of various ML models is provided by this radar plot. Performance patterns that are not apparent in single-metric comparisons are revealed by the multi-dimensional approach. To generate thorough performance profiles, the radar visualisation combines dropout recall, dropout precision, at-risk recall, weighted recall, and weighted F1 scores.

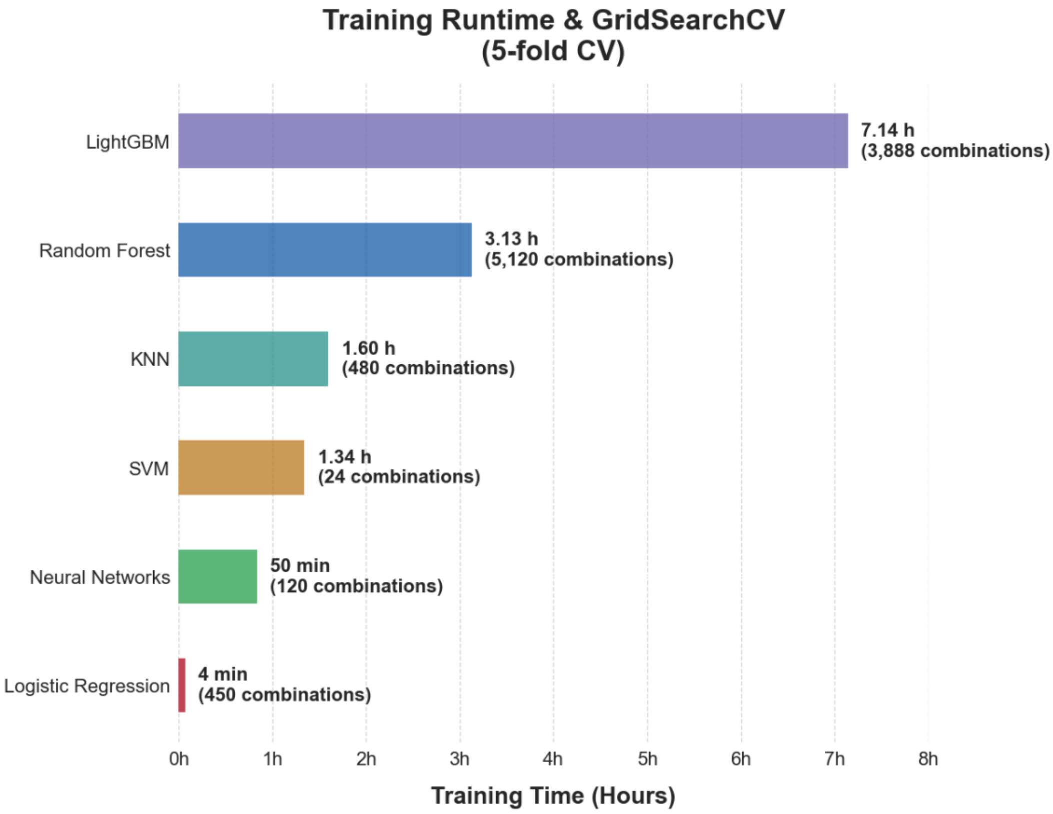

Figure 4.5: Training Runtime and Parameter Combinations Across Models

As previously found, this radar plot visualisation also shows different performance patterns among models. LR is the top performer with the largest radar area, showing balanced strength across several dimensions, including maximum dropout recall (0.668) and consistent performance across recall-focused dimensions. To illustrate why high weighted F1 scores do not ensure minority class identification effectiveness, NN, on the other hand, displays a highly unbalanced radar profile that is concentrated in precision and weighted F1 regions while displaying significant weaknesses in recall dimensions. Performance gradient zones highlight threshold boundaries between excellent (≥ 0.60), good (0.40-0.59), and needs improvement (< 0.40) levels. This allows for quick assessment of model positioning relative to deployment criteria across all evaluation dimensions at once. The radar plot's colour-coded styling distinguishes performance tiers, with excellent performers (LR, SVM) in solid bold styling, good performers (RF, LightGBM) in medium styling, and models needing improvement (KNN, NN) in lighter dashed styling.

A brief summary of the performance metrics for all assessed models is provided in Table 4.1, which combines the in-depth discussion of each machine learning model with the visual analyses that go along with it. Key metrics (at-risk recall, weighted recall, weighted F1 score, dropout recall and precision) and the associated training runtime are highlighted in the table. The dropout recall metric determines the tier classification, as previously established: models with scores above 60% are classified as Top tier, those between 40 and 60% as Mid tier, and those below 40% as Low tier. All things considered, the models can be clearly compared thanks to this summary table, which combines computational effectiveness and predictive performance into a single, understandable picture.

4.7 Best Model - Multinomial Logistic Regression

With a runtime efficiency of 4.1 minutes, the multinomial LR emerged as the victorious machine learning model. Additional information on feature importance, performance metrics, and hyperparameter optimisation is given in this subsection.

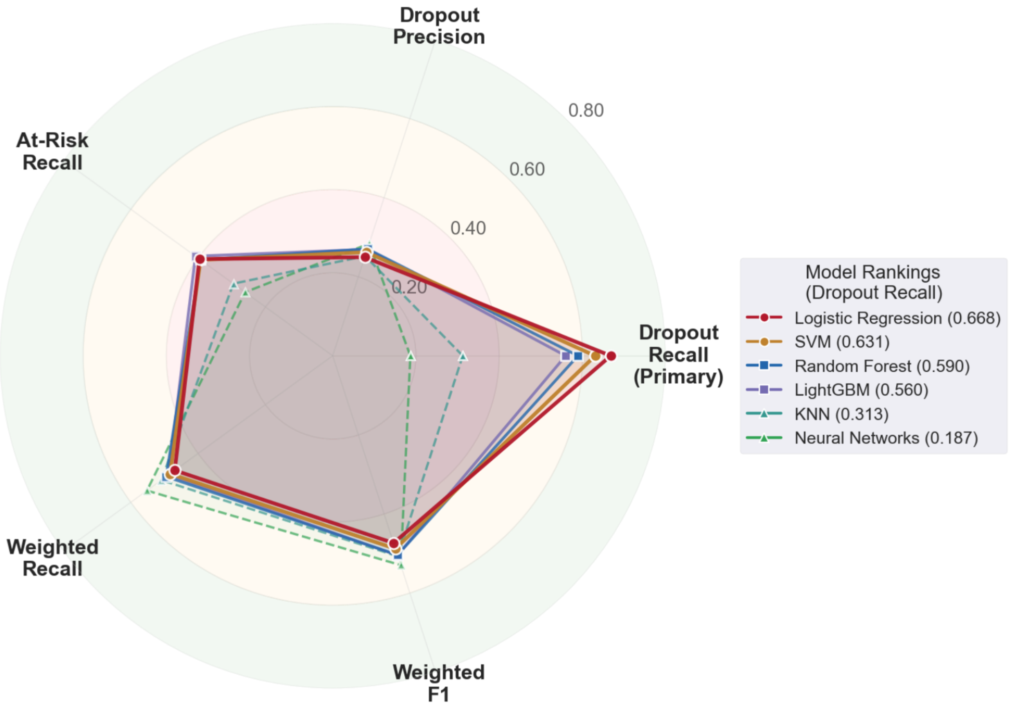

Figure 4.6: Multi-metric Radar Plot for Six ML Models

Rank	Model	Dropout Recall	Dropout Precision	At-Risk Recall	Weighted Recall	Weighted F1	Training Runtime	Tier
#1	LR	0.668	0.250	0.396	0.469	0.474	4m	Top
#2	SVM	0.631	0.262	0.392	0.483	0.488	1.3h	Top
#3	RF	0.590	0.270	0.393	0.495	0.504	3.1h	Mid
#4	LightGBM	0.560	0.269	0.408	0.488	0.505	7.1h	Mid
#5	KNN	0.313	0.252	0.295	0.509	0.507	1.6h	Low
#6	NN	0.187	0.284	0.260	0.552	0.530	50m	Low

Table 4.1: Model Performance Summary Table

4.7.1 Hyperparameter Optimisations

The model predicts three student outcome classes: *Withdrawn* (Class 0), *Fail* (Class 1), and *Pass* (Class 2) using $scikit-learn$'s $LogisticRegression$ class with the $SAGA$ solver and $L1$ regularisation. A thorough GridSearchCV investigated 450 parameter combinations using $5-fold$ cross-validation. Strong regularisation ($C = 0.01$) and custom class weights $0 : 2.09, 1 : 1.31, 2 : 0.48$, which prioritised dropout detection, were features of the ideal setup. The $SAGA$ solver worked best for multinomial classification, and the $L1$ penalty allowed for automatic feature selection while reducing overfitting. To guarantee convergence stability, the maximum number of iterations was set at 1000. Prioritising dropout recall over overall accuracy, the optimisation achieved $66.84\% \pm 3.08\%$ with consistent cross-fold performance, which is in line with early intervention educational goals. Although the top three configurations received identical scores, the chosen model needed the fewest maximum iterations, according to detailed hyperparameter results (Table 4.2). This ensures computational efficiency, which is a critical advantage for larger datasets and real-world deployment for the SDP.

4.7.2 Confusion Matrix and Scoring Metrics Analysis

The model was run using the multinomial LR's top-ranked hyperparameter configuration, and the resulting confusion matrix is displayed in Table 4.3. Performance patterns that are in line with the priorities of educational interventions are revealed by the analysis of this optimal multinomial LR model. The model achieved a recall rate of 65.3%, which closely matches the 66.84% dropout recall seen during cross-validation, correctly identifying 692 cases out of the $1,059$ actual *Withdrawn* students in the test set. 367

Rank	**Score (± std)**	**C**	**l1_ratio**	**max_iter**	**Penalty**	**Solver**
1	0.6684 ± 0.0308	0.01	0.5	1000	l1	saga
2	0.6684 ± 0.0308	0.01	0.5	2000	l1	saga
3	0.6684 ± 0.0308	0.01	0.5	3000	l1	saga
4	0.6122 ± 0.0252	0.01	0.5	1000	elasticnet	saga
5	0.6122 ± 0.0252	0.01	0.5	2000	elasticnet	saga

Table 4.2: Hyperparameter exploration results for Multinomial LR using 5-fold CV

Withdrawn students in total were incorrectly classified (115 predicted as *Fail* and 252 as *Pass*), the latter being the most alarming error type because it suggests no intervention. The model's cautious prediction approach, which was influenced by conscious optimisation decisions, is reflected in this result.

The precision for Class 0 was 26.3%, with 692 correct predictions out of $2,626$ ($692 + 862 + 1,072$) total Class 0 predictions. This indicates that for every correctly identified *Withdrawn* student, roughly three needless interventions were initiated. This resulted in $1,934$ false positives, including $1,072$ *Pass* and 862 *Fail* students who were mistakenly labelled as *Withdrawn*. With the help of the custom class weights $0 : 2.09, 1 : 1.31, 2 : 0.48$, this aggressive identification approach prioritises false positives over false negatives. This deliberate strategy aligns with educational objectives, where it is better to provide unnecessary support than to ignore *Fail* students. As summarised in Table 4.4, with an overall weighted recall of 48.08% and a macro recall of 46.0%, the precision–recall trade-off that results shows how effective the optimisation strategy was. The effectiveness of this unbalanced optimisation design for early intervention systems is confirmed by the purposefully lower precision for Class 0 when compared to Classes 1 and 2.

Actual / Predicted	**0 - Withdrawn**	**1 - Fail**	**2 - Pass**
0 - Withdrawn	692	115	252
1 - Fail	862	224	324
2 - Pass	1072	254	1750

Table 4.3: Confusion Matrix for Multi-class Classification

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0 - Withdrawn	0.26	0.65	0.38	1059
1 - Fail	0.38	0.16	0.22	1410
2 - Pass	0.75	0.57	0.65	3076
Accuracy	0.48			5545
Macro Avg	0.46	0.46	0.42	5545
Weighted Avg	0.56	0.48	0.49	5545

Table 4.4: Detailed Scoring Metrics Report for Multi-class Classification

4.7.3 Feature Importance and Model Interpretability

$L1$ regularisation made it possible to automatically select features using coefficient magnitude analysis. The feature importance ranking that resulted validated carefully crafted variables intended to predict student outcomes and showed that the top 8 features have strong predictive power for dropout detection and distinct patterns of educational relevance.

Top 8 Feature Analysis The top three features, as illustrated in Figure 4.7, are specifically designed variables that exhibit remarkable predictive ability for dropout detection. By capturing academic excellence through composite scoring mechanisms and validating the strategic design approach for student assessment metrics, *Excellent_Score* (coefficient: 0.3293) emerged as the dominant predictor. *Active_in_VLE* (coefficient: 0.2363) came in second with 72% of the top feature's strength, effectively capturing engagement patterns in virtual learning environments that are essential for the success of online education. The engineered variable trilogy was completed by *Student_engagement* (coefficient: 0.1390) at 42% relative strength. It measures comprehensive student participation across multiple dimensions to provide crucial behavioural insights that complement academic performance metrics. Existing demographic and educational variables with significantly lower predictive power made up the remaining five features: Prior educational attainment was captured by *highest_education_ord* (0.0898, 27% relative strength). Accessibility concerns were identified by *disability_binary* (0.0790, 24%), socioeconomic factors were represented by *imd_band_ord* (0.0234, 7%), and the contributions of *age_band_ord* (0.0035) and *region_Wales* (0.0004) were minimal. The significant decrease in coefficients from the lowest engineered variable to the highest demographic variable (0.1390 to 0.0898) reinforces the strategic focus on engineered variables for educational outcome prediction and demonstrates the superior predictive value of purpose-built features over conventional student characteristics.

The multinomial LR provided good interpretability and respectable performance in the larger model comparison framework. The model is appropriate for educational stakeholders who need transparent prediction mechanisms because the linear decision boundaries made it easy to understand feature relationships.

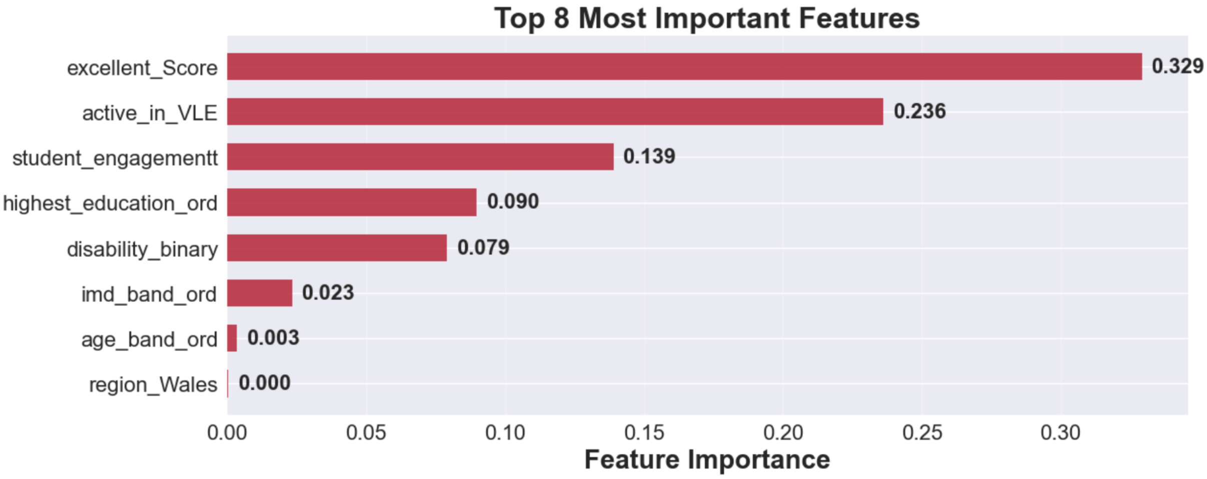

Figure 4.7: Top 8 Most Important Features

Chapter 5

Further Work

The following suggestions provide important avenues for expanding the six-ML model comparison framework in order to improve the robustness and performance of dropout detection.

Enhanced Model Optimisation and Comparison. A methodical, cost-sensitive class-weight search should be used by all models in the current six-ML comparison. In this study, balanced weights (balanced: $0 : 1.7449, 1 : 1.3123, 2 : 0.6006$; custom: $0 : 2.0939, 1 : 1.3123, 2 : 0.4805$) were compared to a single custom weighting scheme. Future research should expand on this strategy by looking for multipliers across a grid for each of the six models. This would directly maximise the $dropout - recall$ objective while maintaining stability across folds and cohorts (for example, multipliers of $+20/0/-20\%$). A structured workshop with academic advisors, educational stakeholders, and domain experts could help pre-narrow the weight space by incorporating domain priors on false-negative costs. These expert-informed choices would then be confirmed by an empirical search across the whole model suite. The Class Weighting Strategy presented in Table 3.6 and Section 3.8 is expanded upon in this procedure. Instead of depending only on ad hoc re-weighting, the six-ML comparison should also assess sampling strategies and principled loss functions in addition to weighting schemes. Modern strategies like Class-Balanced losses, as described in the background (in Section 2.7), provide theoretically sound substitutes; evaluating these in addition to focal loss across all six models would demonstrate whether principled objectives routinely perform better than fixed multipliers. Finally, adopting class-aware resampling or implementing weighted cross-entropy within NN frameworks would restore equitable comparison conditions for models (KNN, NN) that lack native class-weighting mechanisms (in Table 3.6).

Temporal Modelling and Validation. In order to detect dropout risk earlier, the six-ML comparison framework should be expanded to include temporal and survival-style modelling. In order to identify which algorithms best capture behavioural changes over successive assessments and seasonal patterns in student activity, future iterations should evaluate multi-temporal windows across all six models, even though this study limited features to the window before the first assessment (Subsection 3.4.2). Furthermore, the validation strategy needs to be expanded beyond the current 5-fold CV and cohort+outcome stratification split (Section 3.5). A more thorough method would test robustness under cohort shift by methodically evaluating all six models using temporal hold-outs and leave-one-module-presentation-out validation. In order to ensure more equitable comparisons across the algorithm suite, nested CV should also be used to lessen bias in model selection.

Chapter 6

Conclusion

This study created a comparative ML pipeline to anticipate student withdrawal early and facilitate prompt intervention in a virtual learning environment (VLE), as high attrition in VLEs continues to be a persistent challenge. The problem was constructed as a three-class task with *Withdrawn* (Class 0), *Fail* (Class 1), and *Pass* (Class 2) using the OULAD dataset. Through a seven-phase workflow that covered data ingestion to model selection, the goals were to integrate engagement-aware features, socio-demographic signals, and find a workable model for early detection. Early behavioural indicators, such as *excellent_score*, *active_in_VLE*, and a composite *student_engagement* flag, were designed around the first *TMA* window. Custom weights $(0 : 2.09, 1 : 1.31, 2 : 0.48)$ and a dropout-recall-oriented goal were used to address class imbalance. RF, Multinomial LR, KNN, LightGBM, SVM, and NN were the six algorithms that were compared using the same split and scoring criteria. With 19.1% *Withdrawn*, 25.4% *Fail*, and 55.5% *Pass*, the dataset showed an unbalanced class distribution (Table 3.1). The socio-demographic analysis revealed modest gender differences but distinct gradients for age, disability, IMD band, and prior education (Table 3.4), all of which were kept for modelling and interpretation, with the exception of gender.

According to the model comparison, Multinomial LR outperformed SVM at 63.09% and achieved the highest *Withdrawn* recall at 66.84%, exceeding the study's goal of $\geq$ 60%. KNN and NN performed poorly for the minority class, whereas RF and LightGBM formed a mid-tier (Table 4.1). Given its high recall of the class of greatest institutional interest, this ranking supports the choice of LR for production. While SVM took about 1.34h, RF 3.13h, and LightGBM 7.14h for lower or mid-tier recall, the final LR model was trained in 5-fold CV in about 4 minutes, allowing for quick iteration and re-training (Figure 4.4). For regular refreshes and operational use, LR is therefore preferred by the time-to-value profile. The LR confusion matrix revealed the intended bias towards recall in relation to the precision–recall trade-off: *Withdrawn* recall reached 65.3% while precision was 26.3% (Table 4.4). In actuality, more students were flagged than withdrew, which is consistent with early-warning needs where it is more expensive to check for missing *Withdrawn* students than to perform additional checks. This behaviour is also expected under the custom weights and objective.

Engagement variables dominated the LR coefficients in terms of feature importance and interpretation, with *student_engagement* ranking third, *active_in_VLE* second, and *excellent_score* first (Figure 4.7). This demonstrated the high predictive value of straightforward and understandable engagement proxies prior to *TMA1*. Age and regional dummies had little effect on socio-demographics, but *highest_education*, *disability*, and *imd_band* all contributed significant but smaller signals. These results give employees clear explanations, demonstrating that lower withdrawal risk is associated with better early performance and above-average activity. The following is a summary of the research objectives, which were met:

- Engagement features in the models: The three engineered engagement indicators were used and proved to be the best predictors in the winning ML model.

- Socio-demographic associations: Highest education, IMD, and disability all demonstrated significant correlations with outcomes in the quantified and visual relationships; gender was left out because of the lack of practical separation.

- Selecting a useful model: The $\geq 60\%$ recall target was successfully met by LR, which was chosen for its leading recall, short runtime, and interpretability.

Given the modest precision observed, advisors should triage the flagged list of students who are most likely to withdraw after the chosen LR model has been used as a first-line screener (Table 4.4). While thresholds and class weights can be adjusted to local capacity, this process can be supported by straightforward rules like verifying current activity, checking recent *TMA* outcomes, and cross-referencing support history. Since some flagged cases will still result in passing, communication with students should continue to be constructive rather than punitive. OULAD's daily click-log granularity and the UK context set boundaries for external validity in terms of scope and limitations. In order to produce clean "pre-assessment" features, the method also depends on *TMA1* timing assumptions. Additionally, the investigation of alternative weighting or loss strategies and broader temporal modelling were restricted by resource limitations. These limitations influence how the findings are interpreted and how broadly they can be applied to other VLEs.

To conclude, this project provided a clear, socio-economically informed, engagement-aware pipeline for early withdrawal detection. The comparison proved that a regularised multinomial LR can successfully strike a balance between speed, interpretability, and performance, which makes it appropriate for promptly identifying students who are at risk. With human-in-the-loop review, institutions can use it to address the precision trade-off while addressing surface withdrawal risks. For further research, there are opportunities to improve temporal modelling, increase optimisation, and improve fairness controls. The main goal, which was to offer useful and comprehensible predictions that aid in better retention choices in online learning, was ultimately accomplished.

Bibliography

[1] K. Jordan. Massive open online course completion rates revisited: Assessment, length and attrition. *The International Review of Research in Open and Distributed Learning*, 16(3), 2015.

[2] O. Simpson. '22% -can we do better?'-The CWP Retention Literature Review. *ResearchGate*, 2010.

[3] J. Kuzilek, M. Hlosta, and Z. Zdrahal. Open University Learning Analytics dataset. *Scientific Data*, 4(1):170171, 2017.

[4] M. Hussain, W. Zhu, W. Zhang, and S. M. R. Abidi. Student Engagement Predictions in an e-Learning System and Their Impact on Student Course Assessment Scores. *Computational Intelligence and Neuroscience*, 2018(1):6347186, 2018.

[5] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* The MIT Press, first edition, 1992.

[6] R. S. Baker and P. S. Inventado. Educational Data Mining and Learning Analytics. In J. A. Larusson and B. White, editors, *Learning Analytics: From Research to Practice*, pages 61–75. Springer, 2014.

[7] T. Le Quy, A. Roy, V. Iosifidis, W. Zhang, and E. Ntoutsi. A survey on datasets for fairness-aware machine learning. *WIREs Data Mining and Knowledge Discovery*, 12(3):e1452, 2022.

[8] J. A. Martínez-Carrascal, M. Hlosta, and T. Sancho-Vinuesa. Using survival analysis to identify populations of learners at risk of withdrawal: Conceptualization and impact of demographics. *The International Review of Research in Open and Distributed Learning*, 24(1):1–21, 2023.

[9] E. R. Kahu. Framing student engagement in higher education. *Studies in Higher Education*, 38(5):758–773, 2013.

[10] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[11] F. E. Harrell. *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis.* Springer International Publishing, second edition, 2015.

[12] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[13] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[14] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[16] M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

[17] N. Tomasevic, N. Gvozdenovic, and S. Vranes. An overview and comparison of supervised data mining techniques for student exam performance prediction. *Computers & Education*, 143:103676, 2020.

[18] C. Romero and S. Ventura. Data mining in education. *WIREs Data Mining and Knowledge Discovery*, 3(1):12–27, 2013.

[19] D. Gasevic, S. Dawson, T. Rogers, and D. Gasevic. Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success. *The Internet and Higher Education*, 28:68–84, 2016.

[20] O. Viberg, M. Hatakka, O. Bälter, and A. Mavroudi. The current landscape of learning analytics in higher education. *Computers in Human Behavior*, 89:98–110, 2018.

[21] R. Conijn, C. Snijders, A. Kleingeld, and U. Matzat. Predicting student performance from LMS data: A comparison of 17 blended courses using moodle LMS. *IEEE Transactions on Learning Technologies*, 10(1):17–29, 2017.

[22] N. Sclater, A. Peasgood, and J. Mullan. Learning analytics in higher education: A review of UK and international practice, 2016.

[23] M. Hlosta, D. Herrmannova, L. Vachova, J. Kuzilek, Z. Zdrahal, and A. Wolff. Modelling student online behaviour in a virtual learning environment, 2018.

[24] A. A. Nafea, M. Mishlish, A. M. Haban Shaban, M. M. Al-Ani, K. M. A. Alheeti, and H. J. Mohammed. Enhancing student's performance classification using ensemble modeling. *Iraqi Journal for Computer Science and Mathematics*, 4(4), 2023.

[25] L. P. Macfadyen and S. Dawson. Mining lms data to develop an early warning system for educators: A proof of concept. *Computers & Education*, 54(2):588–599, 2010.

[26] The open university grading, n.d. Available at: https://www.scholaro.com/db/Countries/United-Kingdom/Grading-System/The-Open-University-11929.

[27] Y. Liu, S. Fan, S. Xu, A. Sajjanhar, S. Yeom, and Y. Wei. Predicting student performance using clickstream data and machine learning. *Education Sciences*, 13(1):17, 2023.

[28] R. Baker, D. Xu, J. Park, R. Yu, Q. Li, B. Cung, C. Fischer, F. Rodriguez, M. Warschauer, and P. Smyth. The benefits and caveats of using clickstream data to understand student self-regulatory behaviors: Opening the black box of learning processes. *International Journal of Educational Technology in Higher Education*, 17(1):13, 2020.

[29] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[30] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data*, 6(4):15:1–15:21, 2012.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[32] Encoding of categorical variables — scikit-learn course, n.d. Available at: https://inria.github.io/scikit-learn-mooc/python_scripts/03_categorical_pipeline.html.

[33] A. Fernandez, S. Garcia, M. Galar, R. C. Prati, B. , and F. Herrera. *Learning from Imbalanced Data Sets*. Springer International Publishing, 2018.

[34] X.-Y. Liu, J. Wu, and Z.-H. Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2009.

[35] B. Krawczyk. Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[36] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239, 2017.

[37] A. Luque, A. Carrasco, A. Martin, and A. de las Heras. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91:216–231, 2019.

[38] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484, 2012.

[39] M. Grandini, E. Bagli, and G. Visani. Metrics for multi-class classification: An overview, 2020.

[40] L. A. Jeni, J. F. Cohn, and F. De La Torre. Facing imbalanced data: Recommendations for the use of performance metrics. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 245–251, 2013.

[41] T. Tantisripreecha and N. Soonthornphisaj. A novel term weighting scheme for imbalanced text classification. *Informatica*, 46(2), 2022.

[42] J. Opitz. A closer look at classification evaluation metrics and a critical reflection of common evaluation practice. *Transactions of the Association for Computational Linguistics*, 12:820–836, 2024.

[43] P. Akor, G. Enemali, U. Muhammad, R. R. Singh, and H. Larijani. Hierarchical deep learning for comprehensive epileptic seizure analysis: From detection to fine-grained classification. *Information*, 16(7):532, 2025.

[44] M. Orooji and J. Chen. Predicting louisiana public high school dropout through imbalanced learning techniques, 2019.

[45] S. Lee and J. Y. Chung. The machine learning-based dropout early warning system for improving the performance of dropout prediction. *Applied Sciences*, 9(15):3093, 2019.

[46] M. Hlosta, Z. Zdrahal, and J. Zendulka. Ouroboros: Early identification of at-risk students without models based on legacy data. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, pages 6–15. ACM, 2017.

[47] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[48] T. Saito and M. Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE*, 10(3):e0118432, 2015.

[49] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[50] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[51] D. R. Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20(2):215–242, 1958.

[52] J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384, 1972.

[53] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.

[54] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[55] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[56] R. Kohavi. A study of cross validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1137–1145, 1995.

[57] S. Varma and R. Simon. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7(1):91, 2006.

[58] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[59] L. Prechelt. Early stopping—but when? In G. Montavon, G. B. Orr, and K.-R. Muller, editors, *Neural Networks: Tricks of the Trade, Second Edition*, pages 53–67. Springer, 2012.

[60] A. Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the Twenty-first International Conference on Machine Learning (ICML)*, page 78. ACM, 2004.

[61] A. Krogh and J. Hertz. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann, 1991.

Appendix A

Socio-Demographic Visualisations

The linear regression analyses in this appendix look at the correlation between six important socio-demographic factors and student outcomes (*Pass, Fail, Withdrawn*). To comprehend how various demographic factors affect academic performance and dropout patterns, each variable is examined independently. Gender, age band, disability status, highest education level, income deprivation (IMD) band, and region are the six socio-demographic factors that were looked at. The proportionate results across the demographic categories are displayed by fitting three regression lines for each variable; the slopes and R^2 values show the direction and strength of each relationship.

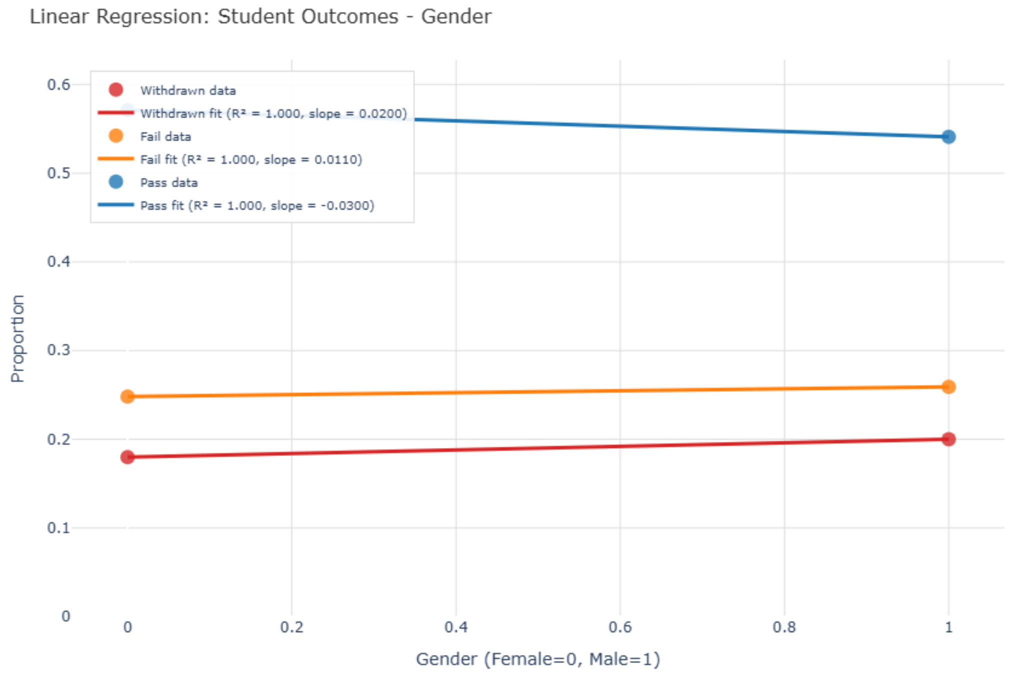

Figure A.1: Student Outcomes by Gender

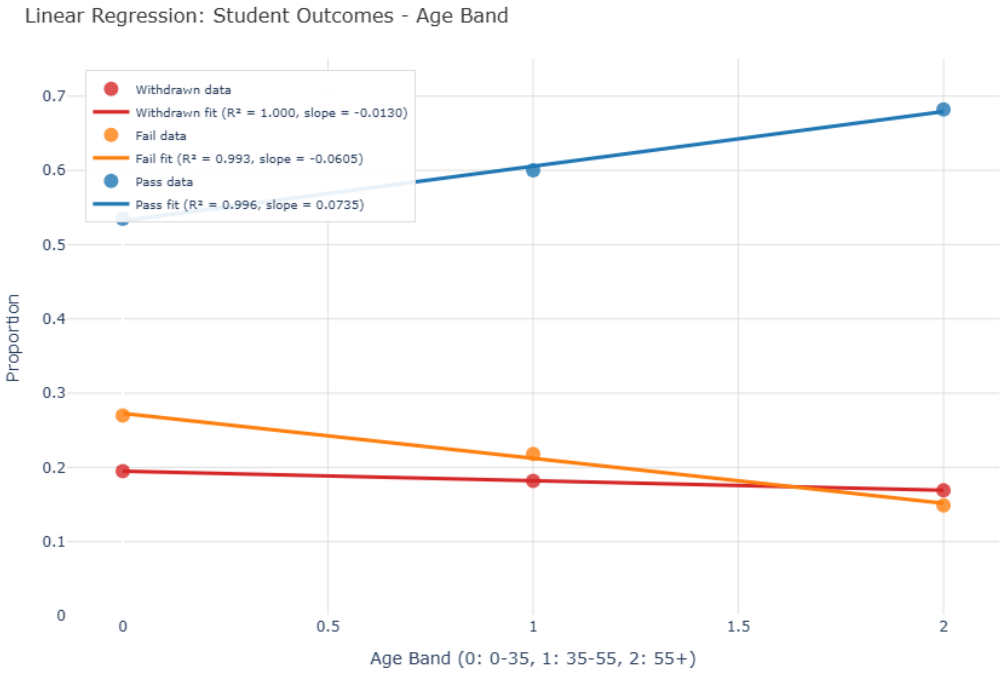

Figure A.2: Student Outcomes by Age Band

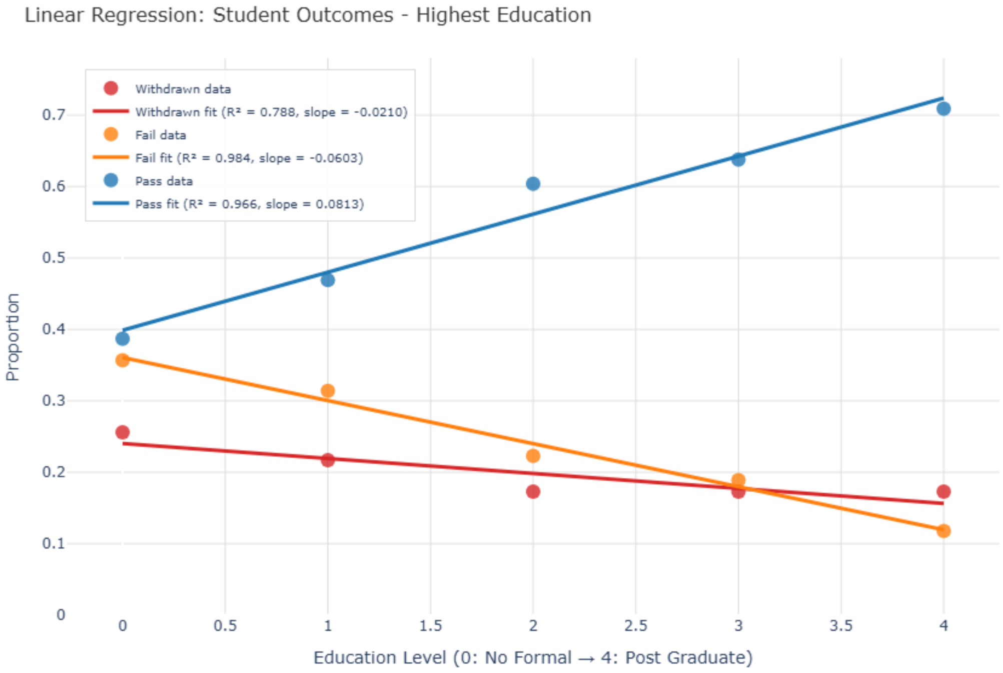

Figure A.3: Student Outcomes by Highest Education Level

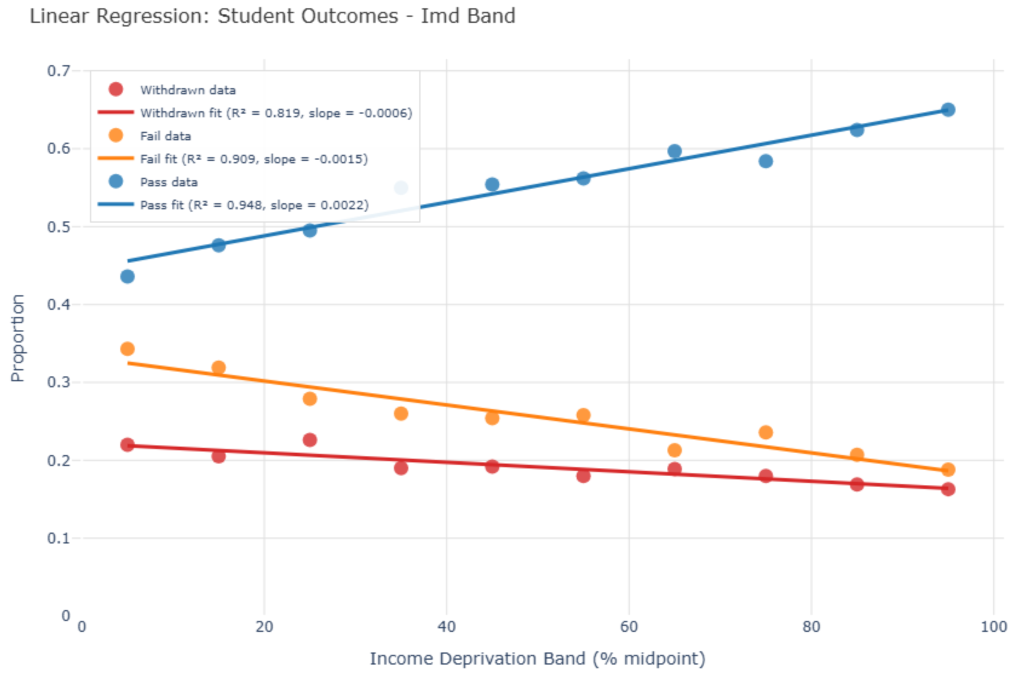

Figure A.4: Student Outcomes by IMD band

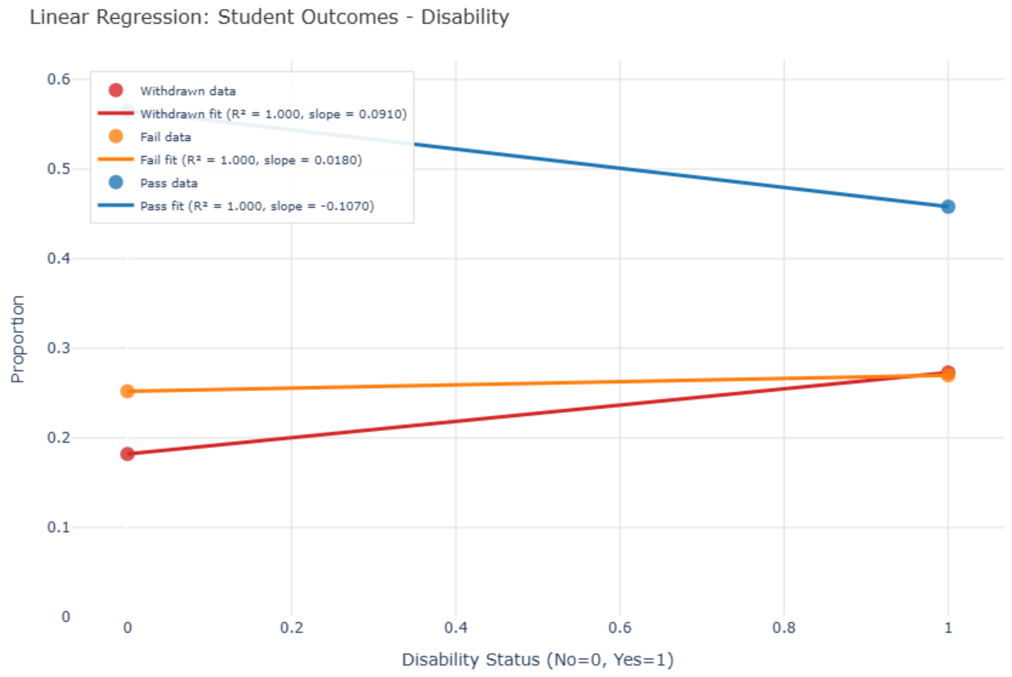

Figure A.5: Student Outcomes by Disability Level

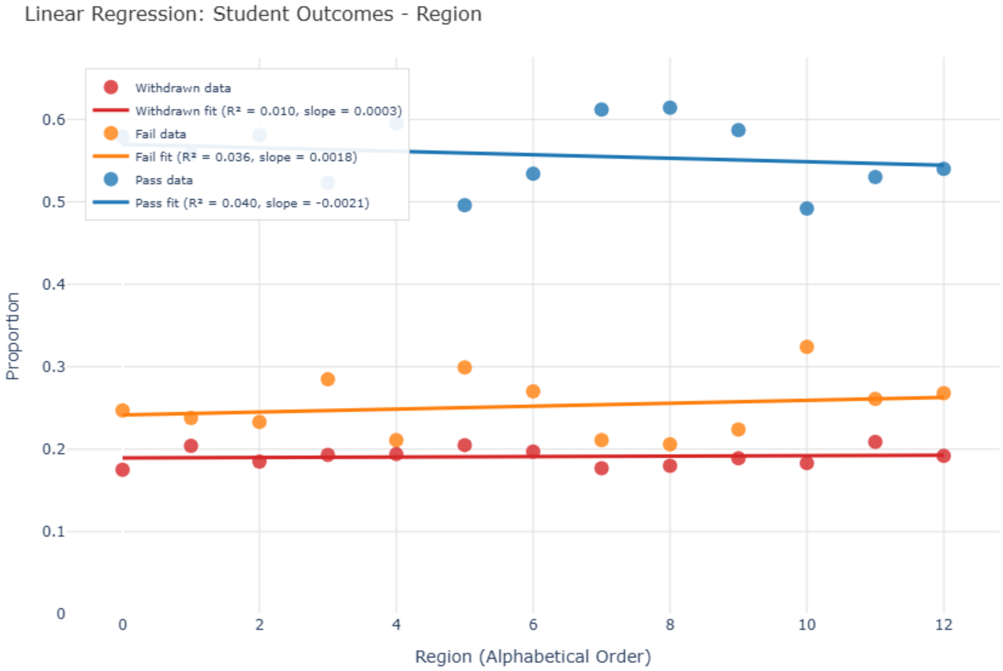

Figure A.6: Student Outcomes by UK Region