UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Manresa School of Engineering

Hierarchical Forecasting for University Enrollment and Student Performance

Master's Thesis

Manresa, May 8, 2026

Master's Thesis (18 ECTS) submmited by

Anass Anhari

in partial fulfillment of the requirements for the

Master's degree in Machine Learning and Cybersecurity for Internet Connected Systems

Advisors: Cristian Maximiliano Rodriguez Rivero & Christoph Bergmeir & Larisa Survilo (Fontys University of Applied Sciences & Universidad de Granada & Riga Technical University)
Counselor: Aleix Llusà Serra

Topics: Memòria TFE; UPC Manresa

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 License. To view a copy of this license, visit `https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode` or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

This manuscript has been typeset using LaTeX and the class `tfgitic` version 7.1

Declaration of Responsibility

I, Anass Anhari, author of this Master's Thesis,

DECLARE

That this project and the accompanying report are original and solely the result of my work. Furthermore, all sources consulted have been included in the bibliography.

By submitting this report for evaluation, I consider this declaration signed for the purposes outlined in the agreement CG/2019/05/10, dated October 8, 2019, of the Governing Council of the Universitat Politècnica de Catalunya, which approves the procedure for plagiarism prevention.

Manresa, May 8, 2026

«From the river to the sea»

For those living through genocide, occupation, and war in Palestine and elsewhere. I hope machine learning and technology research are used for good rather than harm, and that these technologies are always handled responsibly.

Acknowledgments

I want to thank my supervisors, Cristian Rodriguez Rivero, Christoph Bergmeir, and Larisa Survilo, for their help and patience throughout this project. Their advice on forecasting and machine learning helped me work through the harder parts of this research.

I also want to thank the staff at EPSEM-UPC for providing access to the anonymised enrolment records that made this research possible, and to the teams at Riga Technical University (RTU) and Vilnius Gediminas Technical University (VGTU) for their willingness to share data despite the pressures of their own system migrations.

Finally, I want to thank my family for their support and encouragement since *day one*!

Abstract

Accurate prediction of student course enrolment is a relevant problem for resource planning in higher education, yet existing approaches rely largely on aggregate time-series models that ignore individual student trajectories. We investigate how incorporating student-level sequential features into forecasting models affects per-course enrolment prediction accuracy at a Spanish engineering school (EPSEM-UPC, 505 students, 51 courses, 2010–2021). We design a 154-dimensional feature vector encoding enrolment status, normalised grades, cumulative attempt counts, and GPA per academic term, and compare eight modelling approaches: a Naïve persistence baseline, Micro and Macro LSTM networks, decision trees, random forests, XGBOOST, LIGHTGBM, a Course2Vec-enhanced MLP. We also test five hierarchical reconciliation strategies and run an ablation study on the Micro LSTM. Over all 51 courses the Micro model achieves a Mean Absolute Error of 1.73 students per course, close to a strong seasonal Naïve baseline (1.69) that copies the same semester from the previous year. On a filtered core-course subset (excluding predictable first-year and sparse optional courses), the Micro model slightly outperforms the Naïve baseline (2.06 vs 2.07), while a GRU variant achieves the lowest error at 1.64 over all courses. Aggregate models fail to compete due to having only 15 training windows versus 1 322 student-level samples. We document the data-quality challenges we found when trying to collect multi-university enrolment data. Based on this experience, we suggest future directions including hierarchical reconciliation, LMS (Moodle) engagement features, and cross-institutional data harmonisation.

Keywords: enrolment forecasting, deep learning, LSTM, hierarchical forecasting, student profiles, educational data mining, time series.

Resum

La predicció precisa de la matrícula d'estudiants per assignatura és un problema rellevant per a la planificació de recursos a les universitats. Els mètodes existents treballen majoritàriament amb sèries temporals agregades que ignoren les trajectòries individuals dels estudiants. En aquest treball investiguem com la modelització a nivell d'estudiant (mitjançant xarxes LSTM alimentades amb vectors de característiques de 154 dimensions que codifiquen matrícula, notes i intents) afecta la precisió de la previsió de matrícula a l'EPSEM-UPC. Comparem set enfocaments (línia base naïf, LSTM micro i macro, arbres de decisió, boscos aleatoris, XGBOOST, LIGHTGBM, i un MLP amb embeddings de cursos) sobre 505 estudiants i 51 assignatures durant 2010–2021. Sobre les 51 assignatures, el model Micro LSTM obté un MAE de 1.73 estudiants per assignatura, proper a una línia base naïf estacional (1.69). En un subconjunt filtrat d'assignatures troncals (excloent primer curs i optatives), el model Micro supera lleugerament la línia base naïf (2.06 vs 2.07), mentre que una variant GRU aconsegueix l'error més baix (1.64) sobre totes les assignatures. Documentem també les dificultats de qualitat de dades trobades

en l'intent de recollir dades de múltiples universitats, i proposem direccions concretes de treball futur.

Paraules clau: previsió de matrícula, aprenentatge profund, LSTM, previsió jeràrquica, perfils d'estudiants, mineria de dades educatives, sèries temporals.

Contents

Part I.

Memòria

1. Introduction

1.1. Context and Motivation

Higher education institutions must allocate instructors, classrooms, laboratory slots, and teaching assistants well before each new academic term. The quality of these resource-allocation decisions depends on the accuracy of *course enrolment forecasts*, that is, predicting how many students will register for each subject in the upcoming semester. Under-estimating demand leads to overcrowded classrooms and under-staffed courses, while over-estimating it wastes budget on sessions that attract too few participants.

However, course-enrolment forecasting remains difficult in practice because student choices change from year to year, historical patterns break when curricula or policies are updated, and many courses simply do not have enough enrolment history to fit a reliable model. Even when forecasts are accurate, acting on them is not straightforward because staffing, timetabling, classroom availability, accreditation requirements, and budget all impose constraints that a forecast alone cannot resolve. There is also a risk that relying too heavily on forecasts may push small or specialised courses out of the planning process and raise privacy questions when student-level data are involved. It can also narrow academic planning to a pure efficiency exercise at the expense of broader pedagogical goals [Ayg+26].

Enrolment forecasting is harder than it looks. Several factors make it genuinely difficult:

- **Curriculum instability.** Universities add, rename, merge, and retire courses over time. Some institutions are conservative with their offerings while others frequently restructure programmes. A model trained on one curriculum snapshot may not transfer to the next academic year if course codes have changed or entire programmes have been reorganised.

- **Schedule effects.** A student's interest in a course does not guarantee enrolment. Even a student with strong interest in a programming course may choose not to enrol if it is scheduled at six in the morning and creates a four-hour gap in their daily timetable. These constraints are common and hard to model without explicit schedule data.

- **Institutional rules.** Some universities enforce strict prerequisite requirements or cap the number of failed courses before blocking further enrolment. Others give students full freedom to enrol in anything as long as they eventually pass everything. These policies shape enrolment patterns in ways that are difficult to capture without encoding the rules explicitly.

Previous work on enrolment prediction covers statistical time-series models such as ARIMA and exponential smoothing [WK18; LA12], aggregate machine-learning regressors [LEI09; Sha+22], and more recently recommendation-system re-purposing methods [KP24]. The problem with these aggregate approaches is that they model *course demand* at the population level and ignore how each student moves through the curriculum.

The original motivation for this thesis builds on the Bachelor's degree project [Tal23]. The idea was to use individual student trajectories in a *hierarchical forecasting* framework to produce coherent forecasts at the student, course, programme, and institution levels. Reconciliation methods [Hyn+11; WAH19] would ensure that predictions at each level add up consistently. The original plan also included collecting enrolment data from multiple universities to build a cross-institutional forecasting system.

In practice, collecting data from multiple universities turned out to be more difficult than expected. Platform migrations, inconsistent course identifiers, and poorly structured data exports (the default Moodle log exports) from some institutions (*Riga Technical University* and *Vilnius Gediminas Technical University*) made it impossible to build a reliable multi-university training set (see Section 3.1.2 for details). In contrast, the EPSEM-UPC dataset is clean, consistent, and well-structured, which is why it was ultimately used for the analysis. Future work on multi-institution forecasting will need more consistent data formats and clearer course identifiers across institutions before modelling can begin.

We therefore compare several forecasting approaches on the EPSEM-UPC dataset, including persistence baselines, recurrent neural networks, gradient-boosted trees, and learned course embeddings.

1.2. Research Questions

This thesis addresses the following research questions:

RQ How does incorporating individual student academic trajectories into sequential models affect per-course enrolment forecasting accuracy compared to aggregate baselines at EPSEM-UPC?

SRQ1A How does the choice of student-level model architecture (LSTM, GRU, bidirectional LSTM, gradient-boosted trees, embedding-enhanced MLP) affect forecasting performance on small institutional datasets?

SRQ1B How sensitive is the best-performing architecture to hyperparameter choices (hidden size, depth, dropout, window length) in the small-data regime?

SRQ2 How does the number of available training samples at different aggregation levels (student-level vs. term-level) affect the relative performance of Micro and Macro forecasting approaches?

SRQ3 What data-quality and data-acquisition obstacles arise when attempting to build multi-institutional enrolment forecasting datasets, and how do issues such as course-code redundancy and platform migrations constrain the feasibility of cross-institutional modelling?

1.3. Problem Statement

Higher education institutions must make resource-allocation decisions before actual course registrations are fully known, which creates a planning problem that repeats every term. Inaccurate course-enrolment forecasts can lead to overcrowded classrooms, insufficient teaching support, underused facilities, or unnecessary staffing costs. This problem is made worse by fluctuating student choices, curriculum changes, sparse enrolment data for specialised courses,

and institutional constraints such as fixed classroom capacity, instructor availability, and timetabling rules. Better forecasting tools are needed to support planning of instructors, classrooms, laboratory sessions, and teaching assistants while dealing with uncertain student demand.

The goal of this work is to apply hierarchical forecasting methods to academic planning so that forecasts at different levels are consistent with each other. More concretely, this project aims to:

1. Develop an accurate course-enrolment forecasting model that predicts the number of students likely to register for each course before the start of the academic term.
2. Improve academic resource planning by supporting better allocation of instructors, classrooms, laboratory slots, and teaching assistants.
3. Reduce under- and over-estimation of student demand, which in turn reduces overcrowded classes, understaffed courses, and unnecessary operational costs.
4. Account for hierarchical academic structures (faculty, department, programme, year level, course) so that forecasts are consistent across different planning levels.
5. Help universities make better planning decisions based on data, while recognising institutional constraints.
6. Evaluate our forecasting approach and compare it with existing baselines.

Given these challenges, we focus specifically on whether student-level academic trajectories can improve per-course and programme enrolment predictions, as stated in the research questions above.

1.4. Contributions

The main contributions are, ordered by scientific importance:

1. A 154-dimensional multi-hot feature representation of student academic histories that jointly encodes course enrolment, normalised grades, cumulative attempt counts, and GPA per term. This representation makes it possible to model how each student moves through the curriculum over time.
2. An empirical comparison of Micro (student-level) and Macro (aggregate-level) forecasting approaches, showing that student-level modelling works better in small institutional settings because it produces many more training samples from the same observation period.
3. A systematic ablation study showing that adding model capacity beyond a single LSTM/GRU layer leads to overfitting on this dataset, and that the GRU variant outperforms the standard LSTM.
4. An evaluation of five hierarchical reconciliation strategies, showing that reconciliation does not help when the aggregate base forecasts are poor, and identifying when it could become useful.

5. Documentation of data-quality issues from the multi-university data acquisition effort, showing that course-code redundancy, platform migrations, and inconsistent identifiers are the main obstacles to cross-institutional enrolment forecasting.

6. A proposed standardised data format (`*.educast.json`) for institutional enrolment records, designed to reduce the data-harmonisation burden in future cross-institutional studies.

1.5. Thesis Organisation

Chapter 2 surveys related work on enrolment forecasting, sequential student models, hierarchical time-series methods, and learning analytics. Chapter 3 describes the dataset (including the multi-university data acquisition effort), the feature-engineering pipeline, and the model architectures. Chapter 4 describes the experimental setup, preprocessing pipeline, and exploratory data analysis. Results are presented in Chapter 5 and discussed in Chapter 6. Conclusions and future work appear in Chapter 7.

2. Background and Related Work

This chapter covers the background needed to understand the rest of the thesis. We first describe the academic planning problem and how enrolment forecasting fits into it, both at the aggregate and the individual student level. We then review how time-series methods, hierarchical reconciliation, and sequential neural networks have been used in this area. We close by pointing out where prior work falls short (the lack of student-level trajectory modelling, the difficulty of working with small institutional datasets, and recurring data-quality problems), which motivates the approach we take in this thesis.

2.1. The Academic Planning Problem

University resource planning works at two levels. At the *aggregate level*, administrators need to know how many students will enrol in each course so they can assign instructors, book classrooms, open laboratory slots, and hire teaching assistants. At the *individual level*, students decide what to take based on their academic history, which prerequisites they have passed, how much workload they want, and what fits their timetable. The planning problem is that aggregate demand is just the sum of all these individual decisions, but institutions have to commit resources before students have actually registered.

Traditional planning relies on historical averages and manual adjustments by programme coordinators who know their student population informally. This works reasonably well when cohort sizes and curriculum structures are stable, but breaks down when programmes grow, shrink, or restructure their course offerings. As Aygül et al. [Ayg+26] argue, more systematic forecasting approaches can support dynamic course scheduling toward strategic university scaling. They also note that forecasts must be embedded within prescriptive frameworks that respect timetabling, staffing, and capacity constraints.

Rodriguez Rivero et al. [Riv+25] make a related point about AI in education, arguing that emerging technologies should support students' cognitive effort, not replace it. For enrolment forecasting, the same principle applies. The goal is not to predict what students will do with perfect accuracy, but to give planners a better starting point while leaving students free to make their own choices.

The connection between individual student trajectories and aggregate enrolment is naturally hierarchical. Individual students make up courses, courses make up programmes, and programmes make up the institution. This structure is what motivates hierarchical forecasting methods that can produce consistent predictions across all levels at once.

2.2. Related Work

2.2.1. Enrolment Forecasting

Predicting course enrolment requires accounting for past students, new ones, and transfer students whose interests change over time [KP24]. Early approaches used aggregate time-

series methods. Lavilles and Arcilla [LA12] compare simple moving average, single exponential smoothing, and double exponential smoothing to predict total student numbers, incorporating the best-fitting model into their school management system. Wang et al. [Wan+14] introduce a fuzzy time-series model that uses the yearly difference of enrolment as its main domain variable. Watkins and Kaplan [WK18] compare multiple forecasting methods on institutional data and show that Gaussian Process regression performed best in their experiments, beating linear regression, multilayer perceptron, SVM, ARIMA, and exponential smoothing.

Researchers have also applied machine-learning regressors at the course level. Lee et al. [LEI09] propose a modified weighted fuzzy time-series method for enrolment forecasting, while Shao et al. [Sha+22] compare conditional probability analysis, CART, and random forest models for predicting enrolment in individual courses, finding that random forest worked best. Egbo and Bartholomew [EB18] use a multi-layer feed-forward neural network for the same purpose. Aitken et al. [AYM11] provide an institutional perspective and compare simple projection heuristics with richer demographic covariates in an Australian context, finding that student retention data adds predictive value beyond just historical enrolment counts. Loder [Lod25] applies micro cluster learning to predict "active" students, demonstrating that grouping students by behavioural clusters can improve management-level forecasts at Austrian universities. Al-Ahmad et al. [Al-+25] predict academic performance at Saint Cloud State University, showing that ensemble methods achieve high accuracy on student-level prediction tasks even with moderate dataset sizes.

Khan and Polyzou [KP24] propose a re-purposing approach that first generates a *course recommendation set* for each continuing student and then sums the recommendations to estimate per-course enrolment. They test four recommenders (including an LSTM-based model) alongside time-series and regression baselines on a seven-year Florida International University dataset, showing that recommendation-based estimates match or beat direct prediction methods for most courses. However, good individual ranking does not necessarily translate into accurate aggregate enrolment counts. A recommender may correctly order a student's course preferences but assign probabilities that, when summed across the cohort, systematically over- or under-predict demand.

In this thesis, we study the enrolment forecasting problem at EPSEM-UPC by comparing micro-level and macro-level approaches to course enrolment prediction. We also consider the practical constraints of academic data, including course-code redundancy, small cohort sizes, and inconsistencies across institutional sources [Con+17; RV20; Al-+25].

2.2.2. Student Trajectory Modelling

Several groups have modelled individual student progression through a curriculum as a sequential problem, primarily for *course recommendation*. Pardos and Jiang [PJ20] explore course recommendation with a focus on serendipity, comparing skip-gram course embeddings against an existing RNN-based production system at UC Berkeley. Pardos and Nam [PN20] formalise the course embedding approach as Course2Vec, training a Skip-gram model [Mik+13a; Mik+13b] on enrolment sequences to produce dense course representations that reflect curriculum structure. Shao et al. [SGP21] propose PLAN-BERT, a BERT-based model for degree planning that predicts enrolment across multiple consecutive semesters. Polyzou et al. [PNK19] develop *Scholars Walk*, a random-walk model that captures the stochastic transition structure of course sequences.

These methods encode student histories as sequences of multi-hot vectors over the course

catalogue, the same representation we adopt in this thesis. The key difference is the downstream task. Instead of ranking courses for a single student, we use the model's output as a per-student predicted enrolment vector and sum these vectors across all students to obtain course-level demand forecasts. We also evaluate whether reconciling these bottom-up forecasts with aggregate-level predictions can produce *reconciled forecasts* that are consistent across all levels of the hierarchy.

2.2.3. Time-Series Forecasting in Education

Traditional educational enrolment forecasting has often relied on univariate time-series methods that treat per-course enrolment counts as a single series evolving over discrete academic terms. These approaches model the series as a combination of level, trend, seasonality, and noise components.

Moving average and exponential smoothing methods assign fixed or exponentially decaying weights to past observations. Lavilles and Arcilla [LA12] apply these techniques to total student numbers and find single exponential smoothing adequate for short-term projections. *Fuzzy time-series* methods [Wan+14; LEI09] discretise enrolment values into linguistic categories and define transition rules, avoiding the normality assumption of traditional methods. *ARIMA* models capture both autoregressive and moving-average dynamics after differencing the series to achieve stationarity. *Gaussian Process regression* [WK18] provides probabilistic forecasts with uncertainty bands, which Watkins and Kaplan show can outperform parametric methods on institutional data.

More recently, researchers have applied machine-learning regressors to enrolment data. Random forests [Sha+22] and feed-forward neural networks [EB18] treat each term's features (e.g., lagged enrolment, demographics) as a tabular input and predict next-term counts. Loder [Lod25] combines cluster analysis with random forests to identify "active" vs. "inactive" student subpopulations before forecasting. In our earlier Bachelor's thesis [Tal23], we applied decision trees and random forests to per-course enrolment prediction at EPSEM-UPC, finding that depth-4 decision trees achieved the best single-course classification accuracy on the same dataset used here.

These unimodal approaches share several limitations in the university setting:

- They operate on *aggregate counts* and cannot model individual student trajectories or the factors driving each student's enrolment decision.
- The small number of available terms (typically 10–30 for a single institution) limits the temporal evidence available for fitting complex models.
- Series are often *sparse and irregular*, with enrolment patterns disrupted by curriculum changes, course-code renames, and administrative restructuring that break temporal continuity.

These limitations motivate the student-level (Micro) approach adopted in this thesis, which generates many more training samples from the same institutional observation window.

2.2.4. Reconciled Forecasts

Forecast reconciliation is the process of adjusting a set of independently generated forecasts so that they satisfy the aggregation constraints implied by a hierarchical or grouped structure.

Given a hierarchy where bottom-level series (e.g., per-student enrolment) sum to higher-level aggregates (e.g., per-course totals, programme totals), the *base forecasts* produced at each level will generally not be coherent (the bottom-level forecasts will not sum to the top-level forecast). Reconciliation produces *revised forecasts* that are both coherent and, under optimal methods, have lower variance than the base forecasts alone [Hyn+11].

Five main strategies exist:

Bottom-up: Forecast only at the lowest level and sum upward. Simple and assumption-free, but propagates base-level errors into all higher levels.

Top-down: Forecast only at the top level and disaggregate using historical proportions. Relies on the top-level forecast being accurate and the proportions being stable.

Middle-out: Forecast at an intermediate level and use bottom-up below and top-down above. A compromise that can reduce estimation noise at the extremes.

Optimal combination (OLS/WLS): Hyndman et al. [Hyn+11] propose combining all base forecasts through a projection that minimises the trace of the reconciled forecast error covariance under an identity or diagonal covariance assumption.

MinT (Minimum Trace): Wickramasuriya et al. [WAH19] extend the optimal combination by estimating the full base forecast error covariance matrix, producing reconciled forecasts with the minimum possible error variance. The MinT reconciliation matrix is:

$$\mathbf{P} = (\mathbf{S}'\mathbf{W}^{-1}\mathbf{S})^{-1}\mathbf{S}'\mathbf{W}^{-1}, \tag{2.1}$$

where $\mathbf{S}$ is the summing matrix encoding the hierarchy and $\mathbf{W}$ is the covariance matrix of base forecast errors.

Athanasopoulos et al. [Ath+24] provide a detailed review of reconciliation methods, their theoretical properties, and practical performance across domains.

In the university planning context, reconciliation matters because different planning decisions operate at different levels. Classroom booking uses course-level forecasts, staffing plans use programme-level totals, and budget allocation uses institution-level aggregates. Without reconciliation, a department head who sums the course-level forecasts will get a different number than the one produced by the programme-level model, creating inconsistency in the planning process.

For this thesis, reconciliation is relevant because we produce forecasts at two levels (student-level Micro and course-level Macro) and want to combine them coherently. However, MinT requires a reliable estimate of $\mathbf{W}$. This in turn requires a sufficient number of historical forecast errors. With only 15 aggregate-level training windows, the covariance estimate is unstable, and we expect MinT to underperform in our setting. We test this hypothesis empirically in Section 3.4.

2.3. Hierarchical Forecasting in Education

Hierarchical forecasting aims to produce forecasts that are consistent across different aggregation levels. Hyndman et al. [Hyn+11] formalise the problem and propose an optimal combination method that uses GLS regression to produce reconciled forecasts. These forecasts are

unbiased, have minimum variance, and sum consistently across all hierarchy levels. Wickramasuriya et al. [WAH19] extend this with the MinT (Minimum Trace) estimator, which uses the covariance structure of base forecast errors to compute the reconciliation weights. Athanasopoulos et al. [Ath+24], discussed in Section 2.2.4, also review the educational application of these methods.

In the educational context, a natural hierarchy places individual students at the bottom, courses at the next level, programmes above, and the institution at the top (see Figure 2.1). In bottom-up aggregation, individual student forecasts are summed to course-level counts, then to programme and institution totals. This is the approach our Micro model follows. The problem is that bottom-up aggregation propagates base-level errors upward and can amplify them. Reconciliation methods reduce this problem by adjusting base-level forecasts using information from higher aggregation levels.

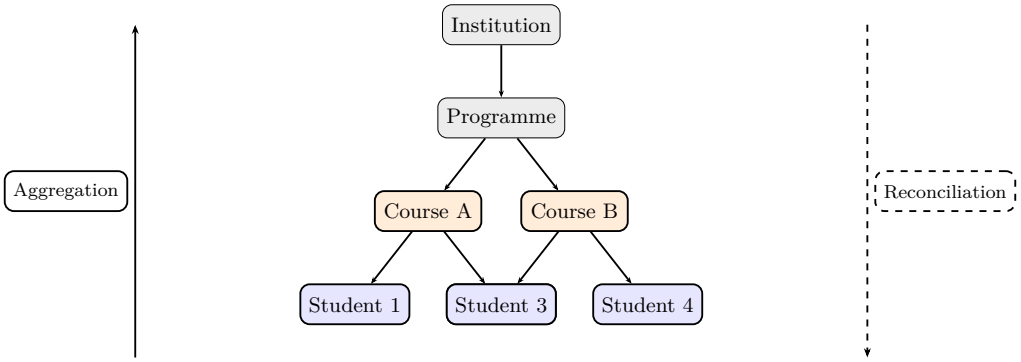


Figure 2.1.: Hierarchical forecasting structure. Aggregation (solid arrow) sums forecasts upward from students to institution. Reconciliation (dashed arrow) adjusts them downward for coherence.

We test five reconciliation strategies in this thesis (Section 3.4), including a simplified MinT-style proportional redistribution. The limited aggregate-level data (15 training terms) prevents the full MinT estimator from improving on the Micro standalone, confirming that reconciliation requires reliable aggregate forecasts to be effective.

2.4. Recurrent Neural Networks

LSTM networks [HS97] are a class of recurrent neural network designed to learn long-range dependencies in sequential data through gated memory cells (see Figure 2.2). The Gated Recurrent Unit (GRU) [Cho+14] simplifies the LSTM architecture by merging the cell state and hidden state into a single vector and replacing the three LSTM gates (input, forget, output) with two (reset and update), reducing the parameter count while retaining comparable performance on many tasks. Researchers have applied both architectures to time-series forecasting [Sal+20] and educational data mining.

These architectures fit student enrolment data well because academic trajectories are irregular. Students progress at different speeds, take semester breaks, repeat failed courses, and sometimes enrol in unexpected combinations. The gating mechanisms of LSTM and GRU cells can handle this because they learn when to keep, update, or discard information from previous time steps. In our setting, we assume that a student's next-term enrolment vector depends on

a short window of previous academic states, including which courses were taken, what grades were obtained, and how many prior attempts were made at each subject.

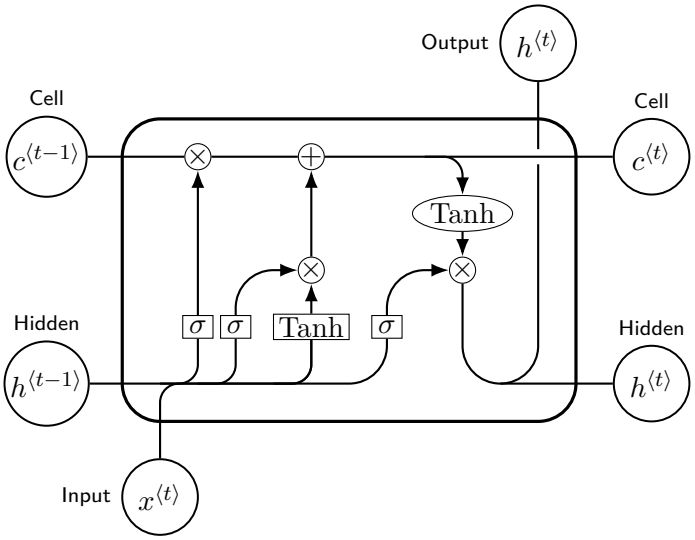


Figure 2.2.: Diagram of an LSTM cell.

In the course-recommendation literature, LSTM-based encoders represent a student's academic history as a fixed-length vector and use it to predict future course preferences [PJ20; KP24]. Our work adapts this approach for enrolment forecasting rather than recommendation. Instead of ranking courses for a single student, we sum predicted enrolment probabilities across all students to estimate course-level demand.

2.5. Gradient-Boosted Trees

Gradient-boosted decision trees have become the go-to method for tabular prediction tasks. XGBOOST [CG16] popularised scalable tree boosting by introducing regularisation (L_1/L_2), support for sparse features, and efficient parallel training. LIGHTGBM [Ke+17] builds on these ideas with a more aggressive leaf-wise growth strategy and histogram-based split finding, which often leads to faster training and lower memory usage, especially on large feature sets.

We include both XGBOOST and LIGHTGBM in our comparison for two reasons. They provide a strong non-sequential baseline for the same prediction task (flattening the 3-term sliding window into a 462-dimensional tabular input), and their built-in feature importance measures help identify which aspects of the student profile carry the most predictive signal.

An important methodological point. In this thesis, tree-based models are used both as *classifiers* (one model per course, trained with a random split, evaluated by F1 score) and as *forecasters* (aggregating predicted probabilities across students, evaluated by MAE on a chronological split). The classification setup uses a random train/test split and treats each course as an independent binary prediction task, which limits direct comparability with the chronological MAE results from the LSTM models. We present both sets of results, but the evaluation protocols are different and the metrics should not be compared across setups without keeping this difference in mind.

2.6. Learning Analytics and Dropout Prediction

A related line of work uses learning management system (LMS) data to predict student performance and dropout risk. Romero and Ventura [RV20] survey the field of educational data mining and learning analytics, covering student performance prediction, dropout detection, and course recommendation as major application areas. Conijn et al. [Con+17] compare 17 Moodle-based blended courses and find that LMS interaction features (login frequency, resource access, forum participation) have limited predictive power for final grades on their own, with explained variance ranging from 8 % to 37 % across courses. Jayaprakash et al. [Jay+14] describe an open-source early alert system that identifies at-risk students from LMS activity patterns.

We include this section even though LMS data was not used in our final experiments. Moodle engagement data was among the signals we tried to incorporate from external institutions (Section 3.1.2), but data quality issues prevented it. The hypothesis linking LMS engagement patterns to student persistence (and therefore to future enrolment decisions) still makes for a promising future direction. A student who stops logging in, submits work late, or misses assignments is more likely to drop out or reduce their course load the following term. Adding such features could improve the Micro model's predictions for students whose enrolment decisions depend more on engagement than on curricular progression alone.

3. Methodology

This chapter describes the dataset, the feature representations, and the models we use to forecast student enrolment. We also document the multi-university data acquisition effort that shaped the scope of this work.

Research design. This study is a retrospective, observational, offline forecasting evaluation with no causal claims. We train models on historical enrolment records and evaluate them on a held-out chronological test period (2018–2021). The study design is purely predictive. We measure whether models can forecast enrolment counts accurately, not whether any intervention changes enrolment behaviour. Across all model comparisons, we control for the dataset (same EPSEM-UPC records), the target definition (per-course enrolment counts aggregated from student-level predictions), the evaluation horizon (one term ahead), and the test period (2018–2021). For the forecasting models, we use a chronological split to avoid temporal leakage. For the classification models, we use a random split (as described in Section 4.3), which means their results are not directly comparable with the forecasting MAE values.

3.1. Dataset

3.1.1. EPSEM-UPC Enrolment Records

The primary dataset was provided by the Escola Politècnica Superior d'Enginyeria de Manresa (EPSEM-UPC) and contains anonymised course registration records for the Enginyeria de Sistemes TIC degree programme. Table 3.1 summarises its main characteristics.

Property	Value
Students	505
Courses	51 (49 with enrolments)
Enrolment records	11 928
Academic years	2010–2021
Terms per year	2 (autumn, spring)
Total terms	23
Missing grades	155 (1.3 %)
Overall pass rate	≈ 70 %

Table 3.1.: Summary of the EPSEM-UPC TIC dataset.

Each record encodes a student identifier, an academic term (year and semester), a course code, the grade obtained on a 0–10 scale (if available), and the cumulative number of prior attempts at the same course. Students who are still enrolled but have not yet received a grade appear with a missing grade value. We encode these as -1 in the feature representation (Section 3.2).

Course popularity is highly imbalanced. First-year compulsory subjects (Matemàtiques Bàsiques, Física, Informàtica, etc.) appear in nearly every student's record, while advanced electives attract fewer than 50 enrolments across the entire dataset. This imbalance directly affects model evaluation, since per-course MAE is heavily influenced by courses with small and highly variable enrolment counts.

Student inclusion criteria. The dataset includes all 505 students who registered for at least one course in the Enginyeria de Sistemes TIC programme between 2010 and 2021. We define a *continuing student* as any student who appears in at least two distinct academic terms. First-year entrants who enrol in term 1 and never return (approximately 45 students with only 1 term of activity) are included in the population but cannot contribute training samples for the LSTM models because the sliding-window mechanism requires a minimum of 4 enrolled terms (window size 3 plus one target term). Students with fewer than 4 terms are excluded *only from LSTM training*. They remain in the aggregate counts used for evaluation and in the tree-based models that do not require temporal windows.

Representativeness and limitations. This dataset covers a single engineering programme at a single institution. The results are internally useful for planning at EPSEM-UPC but have limited external generalisability. The 51 course codes include all courses offered to the programme, whether currently active or historical. Some codes may refer to renamed or reorganised versions of the same underlying subject (see Section 3.1.4), which inflates the apparent course catalogue size and introduces noise into per-course MAE calculations.

3.1.2. Data Acquisition and Institutional Collaboration

The original plan for this thesis, proposed in the Bachelor's degree project [Tal23] (building on the earlier prototype in [Tal23]), was to collect enrolment data from multiple universities and build a hierarchical forecasting system that could operate across institutions. We contacted several European universities and obtained data exports from two of them:

- **Riga Technical University (rtu), Latvia** provided Moodle platform logs and enrolment records for several degree programmes.
- **Vilnius Gediminas Technical University (vilnius tech), Lithuania** also shared Moodle logs and enrolment data for one of their engineering programmes.

The Moodle data from RTU contained useful indicators of student behaviour such as login frequency, time elapsed between task publication and submission, resource access patterns, and session durations. We hypothesised that these engagement metrics could serve as predictors of student persistence. A student who logs in regularly and submits assignments on time is likely more committed to continuing their studies. This signal could complement the enrolment-based features and feed into a dropout risk component of the forecasting pipeline.

We spent a lot of time cleaning these external datasets, but three main issues prevented us from using them for model training:

1. **Platform migrations.** Both institutions had undergone internal migrations in the years preceding the data export. The exported records reflected the messy state of these transitions. Data that should have followed a clean relational schema was instead a patchwork of old and new formats, with fields renamed, restructured, or missing entirely.

2. **Course-code inconsistency.** Course identifiers were neither unique nor stable across academic years. The same course appeared under multiple codes, and different courses sometimes shared codes after a migration. We spent considerable time building course-code mappings through co-enrolment graph analysis and recurrence analysis (tracking which codes replaced others across years). Despite these efforts, the mappings could not be validated with sufficient confidence to use for model training.

3. **Unstructured exports.** The data exports contained large volumes of unrelated records, no clear schema documentation, and duplicated entries. The IT teams at the participating institutions were working under their own migration deadlines and could not dedicate the time needed to produce clean, research-grade exports.

We are grateful to RTU and VILNIUS TECH for their willingness to share data and for the time their teams invested in preparing the exports. Internal migrations are complex projects with their own priorities, and producing well-documented data exports for external research is extra work that is not always possible. We learned that real-world educational data is messy, and any future multi-institution study will need to invest heavily in data harmonisation before modelling can begin.

As an example of our attempts to salvage useful structure from the heterogeneous exports, we built a course co-enrolment graph where nodes are courses and edges connect pairs of courses frequently taken by the same students. We then applied community detection to extract densely connected "programme-like" clusters. Figure 3.1 shows one such cluster (ID 10). While visually informative, these clusters were not stable enough to serve as a ground-truth mapping between programme structures and course identifiers. Without reliable metadata and stable IDs across years, the graph can merge courses that are merely co-taken in common study paths and split programmes whose codes were changed by migrations.

We therefore focused the experimental work on the EPSEM-UPC dataset, where data quality could be verified directly through collaboration with the school's administration.

Connection to SRQ3. The data-acquisition experience described above is our answer to SRQ3. The main obstacles to multi-institutional modelling turned out to be about data, not algorithms. Course codes change when platforms are migrated, there are no stable join keys between systems, and no agreed-upon export schemas exist. We tried several preprocessing strategies (co-enrolment graph clustering, recurrence analysis of course codes, manual inspection of syllabi), but none produced mappings reliable enough for model training. The course co-enrolment graph (Figure 3.1) showed qualitative programme structure but could not be validated as ground truth because the institutional metadata was either missing or inconsistent. In short, cross-institutional enrolment forecasting is a data-harmonisation problem before it is a modelling problem.

3.1.3. The Educast Data Model

To handle enrolment data from different institutions in a uniform way, we developed a standardised data format called `*.educast.json`. The format is defined as a Pydantic schema (Python) with five entity types organised hierarchically:

- **University:** root aggregate containing departments, programmes, and students.

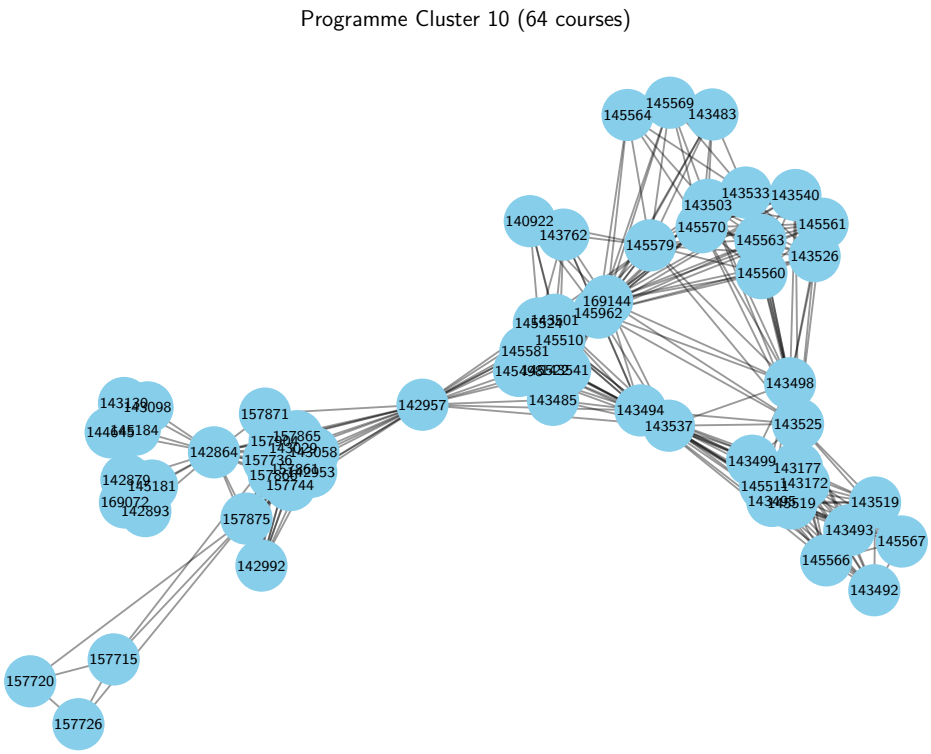


Figure 3.1.: Example community extracted from a course co-enrolment graph (cluster ID 10). Nodes are course identifiers and edges indicate frequent co-enrolment by the same students. The plot illustrates how graph clustering can provide qualitative insight into curriculum structure, but also why it cannot replace stable identifiers and curated metadata when harmonising multi-university exports.

- **Department / Programme:** organisational units that group courses.
- **Course:** an offered subject with an identifier, name, acronym, and credit count.
- **Student:** an individual with demographic metadata (birth year, access pathway, access grade) and an academic history.
- **AttemptedCourse:** a student's attempt at a course in a given term, including grade, semester number, scholarship status, and optionally a list of LMS clickstream events.

The following Python snippet shows the core of the schema:

```
class AttemptedCourse(BaseModel):
    course: Course
    year: Optional[int]
    term: Optional[int]
    grade: Optional[float]
    clickstream: List[ClickstreamData] = []
```

```
class Student(BaseModel):
    student_id: str
    history: AcademicHistory
    access_grade: Optional[float]

class University(BaseModel):
    name: str
    departments: List[Department]
    programmes: List[Programme]
    students: List[Student]
```

This schema supports universities with different term structures (semesters, trimesters, quadrimesters), different grading scales, and optional clickstream data. Table 3.2 shows how the raw EPSEM CSV fields map to the schema fields, and indicates which fields are mandatory for the forecasting pipeline.

Raw CSV Field	Schema Field	Required	Notes
`id_alumne`	`student_id`	Yes	Anonymised hash
`codi_assig`	`course.course_id`	Yes	6-digit code
`nom_assig`	`course.name`	No	Display only
`curs_academic`	`year`	Yes	Academic year
`quadrimestre`	`term`	Yes	1 or 2
`nota`	`grade`	No	0–10 scale
`convocatoria`	`semester` (attempt)	No	Exam sitting
—	`clickstream`	No	LMS data (unused)
`via_acces`	`access_pathway`	No	Metadata
`nota_acces`	`access_grade`	No	0–14 scale

Table 3.2.: Field mapping from raw EPSEM CSV exports to the `*.educast.json` schema. "Required" indicates whether the field is mandatory for the forecasting pipeline to function.

The raw CSV enrolment records from EPSEM are converted to this format through an ETL pipeline (`build_json.py`), which also enriches the data with course acronyms and student metadata from auxiliary files. All downstream processing (feature engineering, model training, evaluation) reads from the `*.educast.json` file, so all downstream processing uses the same data.

The schema was validated on the EPSEM-UPC dataset and partially tested against the structure of the RTU and VILNIUS TECH exports. The external datasets could be parsed into the schema structure, but the quality issues described above (missing fields, inconsistent identifiers) prevented using them for training. We also built an interactive application around this format (Section 6.5).

3.1.4. Data-Quality Issues

Exploratory analysis of the EPSEM dataset revealed two main obstacles to reliable modelling. We describe each issue and explain how it affects model error.

1. **Course-code redundancy.** The number of unique course codes offered per year shows unexplained variance. We observe 51 distinct codes in the full catalogue, of which 49

have actual enrolment records. Of these, approximately 8–10 codes appear for only 1–2 academic years before disappearing, while a new code appears in the same curricular slot. These are suspected renames or reorganisations of the same underlying subject, but we could not confirm this reliably. This affects model error in two ways. First, course-code redundancy *increases apparent sparsity* (a course that has been taught for 10 years appears to have only 3–4 years of history under any single code). Second, it *weakens temporal continuity* (the model sees a "new" course appearing with no history, preventing it from learning enrolment patterns).

2. **Irregular student profiles.** While some students follow the expected progression (taking courses in the prescribed semester order and passing on the first attempt), many show irregular trajectories with failed courses, multiple retakes, semester gaps, and non-standard course combinations. The median student is active for 8 terms (4 academic years), but approximately 45 students ($\approx 9\,\%$) have only 1–2 terms of activity (likely dropouts or early transfers) and are excluded from LSTM training by the windowing requirement. A small group of approximately 30 students extends beyond 12 terms (6+ academic years). The expected effect is that short-history students contribute no sequential training signal, and long-history students with many retakes may introduce noise, since the model might treat repeated courses as a general progression pattern when they are not.

3.2. Feature Engineering

We use three complementary feature representations, each suited to a different family of models.

3.2.1. Multi-Hot Feature Vector

For every student s and academic term T_i, we construct a 154-dimensional real-valued vector. Let $\mathcal{C} = \{C_1, \ldots, C_{51}\}$ be the set of courses in the curriculum. The feature vector $\mathbf{x}_i \in \mathbb{R}^{154}$ is defined as:

$$\mathbf{x}_i = [\, E_i \mid G_i \mid A_i \mid \mathrm{GPA}_i \,], \tag{3.1}$$

where:

- $E_i \in \{0,1\}^{51}$ is the *enrolment* sub-vector: $E_{i,k} = 1$ if student s was enrolled in course C_k during term T_i, and 0 otherwise.
- $G_i \in [-1,1]^{51}$ is the *grade* sub-vector: $G_{i,k} = \mathrm{grade}_{i,k}/10$ if C_k was taken, and $G_{i,k} = -1$ as a sentinel for courses not taken.
- $A_i \in \{0,1,\ldots,5\}^{51}$ is the *attempts* sub-vector: $A_{i,k}$ counts the cumulative number of prior attempts at course C_k, capped at 5.
- $\mathrm{GPA}_i \in [0,1]$ is the cumulative grade point average across all passed courses up to term T_i.

Justification of the -1 sentinel. We encode "course not taken" as $G_{i,k} = -1$ rather than 0 because 0 is a valid (minimum) grade on the 0–10 scale. The sentinel must be distinguishable from any true grade value. We chose -1 for three reasons. First, it lies outside the normalised grade range $[0, 1]$, making it unambiguous. Second, sigmoid and ReLU activation functions in the downstream network can learn to treat negative inputs differently from positive ones. Third, the alternative of masking (setting not-taken positions to zero and using a separate binary mask channel) would double the grade sub-vector dimensionality. We did not run a formal ablation comparing -1 against masking or against grade removal, which remains a direction for future work. The model may interpret -1 as a numeric value during early training, but the LSTM's gating mechanism and the projection layer should learn to associate it with the "not enrolled" state, particularly because the enrolment sub-vector E_i already provides explicit enrolment information.

The full student history up to term T_{n-1} is represented as the sequence $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n-1})$. For the LSTM models, we extract sliding windows of three consecutive terms, producing input tensors of shape $(3, 154)$. This window size balances temporal context against the number of usable training samples, since students with fewer than four enrolled terms contribute no training windows and are excluded.

3.2.2. Tabular Features

For the tree-based classifiers (decision trees, random forests, XGBOOST, LIGHTGBM), we use the tabular encoding developed in the TFG [Tal23]. Two transformations are defined:

- **Transform 1 (T1):** For each student snapshot at term T_i, the feature vector contains binary enrolment flags for every course in every prior term. This produces a wide, sparse matrix (up to 1 035 columns) that grows with the student's history length.
- **Transform 2 (T2):** A more compact encoding that adds normalised grades and attempt counts to the binary enrolment flags, along with metadata such as access pathway, enrolment order, and scholarship status. This gives 159 columns.

In both cases, we train one classifier per course. The target for each model is a binary label indicating whether the student will enrol in that particular course in the next term.

A note on comparability. The T1/T2 tabular models are *classification* models evaluated with a random train/test split and assessed by precision, recall, and F1. The Micro and Macro LSTM models are *forecasting* models evaluated with a chronological split and assessed by MAE on aggregate counts. These two approaches differ in target definition (binary per-student vs. count per-course), split strategy (random vs. temporal), and metric (F1 vs. MAE). We report both because each answers a different question. The classification results show how well we can predict individual enrolment decisions, while the MAE results show how well aggregate forecasts serve resource planning. But the reader should not compare F1 scores with MAE values directly.

3.2.3. Course Embeddings

We learn dense course representations by treating enrolment histories as sentences and applying the Skip-gram algorithm [Mik+13b] with negative sampling [Mik+13a]. For each student, we

construct two types of sentences. The first groups all courses taken in the same semester (capturing co-enrolment), and the second concatenates the entire chronological course history (capturing prerequisite relationships). We train a 16-dimensional embedding on the resulting 3 042 sentences with a context window of 5.

The embedding corpus consists of 3 042 sentences derived from students in the *training period only* (pre-2018). We do not include test-period enrolment sequences in the embedding training corpus to avoid data leakage. The resulting embeddings are frozen and applied identically to both training and test samples.

The trained embeddings encode curriculum structure. Courses from the same semester cluster together in embedding space, and courses linked by prerequisite chains appear as nearest neighbours. We use these embeddings to augment the multi-hot features with three derived signals per time step. These are the mean embedding of enrolled courses, a grade-weighted mean embedding, and a trajectory vector (the difference between consecutive mean embeddings). This produces a 574-dimensional enhanced feature vector ($462 + 112$) that we feed to a feedforward MLP.

3.3. Models

We evaluate models from four families. These are a persistence baseline, recurrent neural networks operating at different aggregation levels, tree-based classifiers, and a feedforward network augmented with learned course embeddings. This range of models lets us compare whether sequential modelling, tabular classification, or representation learning works best for enrolment prediction. Table 3.3 provides an overview of all models, their inputs, targets, and the research questions they address.

Model	Role	Input	Target	Loss	Split	RQ
Naïve	Baseline	Lag-2 counts	Count/course	—	Chrono	RQ
Micro LSTM	Seq. micro	154-dim × 3	Binary/course	BCE	Chrono	SRQ1a
Macro LSTM	Seq. macro	204-dim × 3	Count/course	MSE	Chrono	SRQ2
Decision Tree	Tab. baseline	T1/T2 tabular	Binary/course	Gini	Random	SRQ1a
Random Forest	Tab. baseline	T1/T2 tabular	Binary/course	Gini	Random	SRQ1a
XGBoost	Boosted tab.	T2 tabular	Binary/course	Log	Random	SRQ1a
LightGBM	Boosted tab.	T2 or 462-dim	Binary/course	Log	Both	SRQ1a
Course2Vec MLP	Emb. enhanced	574-dim flat	Binary/course	BCE	Chrono	SRQ1a

Table 3.3.: Model comparison matrix. "Chrono" = chronological split, "Random" = random 80/20 split, "Both" = evaluated under both protocols. All forecasting models produce per-course counts by summing predicted probabilities across students.

3.3.1. Naïve Baseline

We start with the simplest possible predictor to set a baseline. The Naïve baseline simply predicts the enrolment count for each course in term T as the observed count from the same semester of the previous year (lag-2, since the academic calendar has two terms per year). In effect, it assumes that the autumn 2019 enrolment for each course will be identical to the autumn 2018 enrolment.

We chose lag-2 over lag-1 (previous term) because university enrolment exhibits strong annual periodicity. Autumn and spring semesters have different course offerings (some subjects are only taught in one semester), so copying last term's counts would systematically predict zero for courses that alternate between semesters. The lag-2 predictor captures this seasonal structure for free.

This baseline is stronger than it looks. It requires no training, no feature engineering, and no hyperparameter tuning, yet it captures the dominant pattern in the data. Enrolment at EPSEM is stable year over year, and most of the variance comes from slowly-changing cohort sizes, not from sudden shifts in student preferences. Any learned model needs to beat this persistence pattern to be useful. As we show in Chapter 5, the Naïve baseline achieves an MAE of 1.69 students per course over all 51 courses, making it a surprisingly strong benchmark. Only the best learned models achieve small improvements, while aggregate-only models remain far worse. For clarity, the Naïve baseline must be computed on the same evaluation target as the learned models (course counts aggregated from the same student-level windows). Using mismatched aggregate tables inflates the reported error and leads to invalid comparisons.

3.3.2. Micro LSTM (Student-Profile Model)

The Micro model treats each student as an independent sequence. A projection layer maps the 154-dimensional input to a 64-dimensional intermediate representation at each time step. An LSTM encoder [HS97] with 128 hidden units reads the projected sequence and produces a context vector from its final hidden state. Two dense layers (64 units with RELU, then 51 units with sigmoid) map this context to per-course enrolment probabilities.

Per-course enrolment counts are obtained by summing the predicted probabilities across all continuing students in a given term. The model is trained with binary cross-entropy loss, treating each of the 51 courses as an independent binary classification task.

3.3.3. Macro LSTM (Aggregate Model)

The Macro model operates at the course level. The input at each time step is a vector of aggregate statistics per course (enrolment count, mean grade, number of passes, and number of failures), yielding a 204-dimensional input (4×51). An LSTM with 128 hidden units reads a window of three consecutive terms and predicts the 51-dimensional enrolment count vector for the next term, trained with MSE loss.

The main problem with the Macro model is that it has very little training data. Each training sample corresponds to one academic term, where the entire cohort is collapsed into a single 204-dimensional observation. With only 23 terms in the dataset, we obtain at most 15 training windows (after subtracting the window size and the test period). This is 88 times fewer samples than the Micro model's 1 322 training windows. The Micro model avoids this bottleneck because it generates one sample *per student per eligible term*, yielding approximately 2 158 total samples (505 students $\times \approx 4$ eligible windows each), of which 1 322 fall into the training period. This asymmetry is why the Macro model performs poorly, and it affects any aggregate time-series approach when the institutional history is short.

3.3.4. Decision Trees and Random Forests

Following the approach from the TFG [Tal23], we train one decision tree and one random forest per course. The decision tree uses a maximum depth of 4 (depth 5 was found to overfit in the

TFG experiments). The random forest uses an automatically selected number of estimators per course. Both are trained on the tabular T1 and T2 feature representations described in Section 3.2.2.

3.3.5. XGBoost

We train one XGBOOST classifier [CG16] per course using the T2 features. XGBOOST adds L_1 and L_2 regularisation to the boosted tree framework and handles class imbalance through its `scale_pos_weight` parameter, which we set to the negative-to-positive ratio for each course. We use 200 boosting rounds, a maximum tree depth of 4, and a learning rate of 0.1.

3.3.6. LightGBM

LIGHTGBM [Ke+17] uses leaf-wise tree growth and histogram-based split finding, which tends to converge faster than the level-wise strategy of classical gradient boosting. We train one LIGHTGBM classifier per course on two different feature representations:

- On the tabular T2 features (159 columns), for direct comparison with the other tree-based classifiers.
- On the flattened multi-hot windows ($3 \times 154 = 462$ columns), for direct comparison with the Micro LSTM. In this setting, the model receives the same input as the LSTM but without sequential processing, and we aggregate predicted probabilities across students to obtain course-level counts.

3.3.7. Course2Vec MLP

The Course2Vec MLP combines the flattened multi-hot features (462 dimensions) with the embedding-derived features described in Section 3.2.3 (112 dimensions), for a total of 574 input features. The network has two hidden layers (128 and 64 units) with RELU activations, dropout (0.3 and 0.2), and a 51-unit sigmoid output. It is trained with binary cross-entropy loss for 100 epochs.

3.4. Hierarchical Reconciliation

The Micro and Macro models produce forecasts at different levels of the student–course hierarchy. In a fully coherent hierarchical forecasting framework [Hyn+11; WAH19], base-level forecasts (individual students) would be aggregated bottom-up and reconciled with top-level forecasts (course totals) to ensure additivity.

Figure 3.2 illustrates the hierarchical structure and shows which levels are addressed by each model in this thesis.

The general reconciliation problem, as formalised by Wickramasuriya et al. [WAH19], seeks a matrix $\mathbf{P}$ such that the reconciled forecasts $\tilde{\mathbf{y}} = \mathbf{SP}\hat{\mathbf{y}}$ are coherent (they sum correctly across levels) and optimal (they minimise the trace of the forecast error covariance). Here $\mathbf{S}$ is the summing matrix that encodes the hierarchical structure and $\hat{\mathbf{y}}$ is the vector of base forecasts at all levels. The MinT estimator uses the sample covariance of base forecast errors to compute $\mathbf{P}$.

Figure 3.3 illustrates the reconciliation idea for our two-level case.

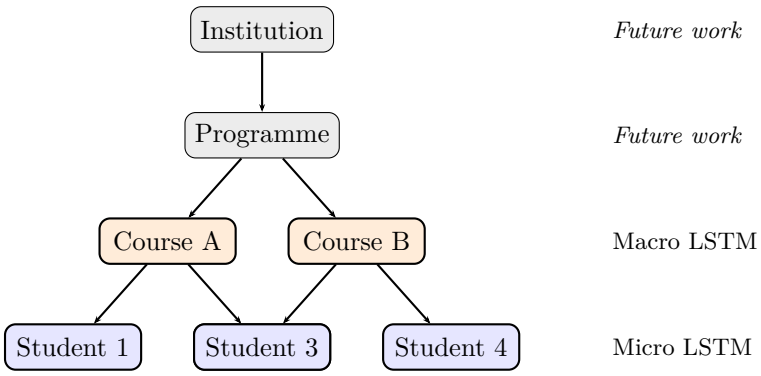


Figure 3.2.: Hierarchical forecasting structure. Bottom-up aggregation sums student-level forecasts to course level, then to programme and institution. This thesis addresses the student (Micro) and course (Macro) levels. Programme and institution levels remain for future work.

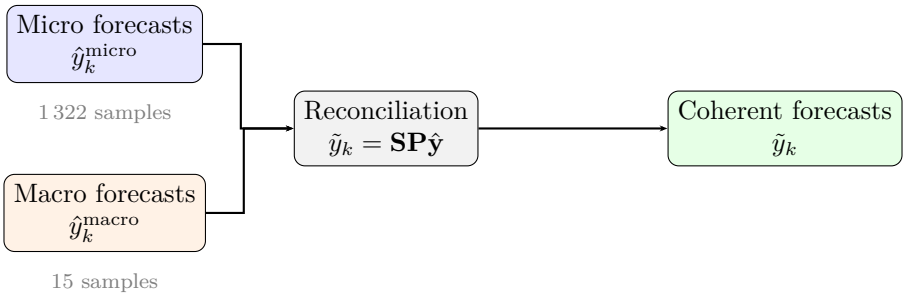


Figure 3.3.: MinT reconciliation concept. Base forecasts from the Micro and Macro models are combined through a projection matrix $\mathbf{P}$ that ensures coherent, variance-minimising forecasts. The sample size imbalance (1 322 vs 15) limits the quality of the Macro input, which in turn limits reconciliation gains.

In our simplified two-level setting (student level and course level), reconciliation reduces to a weighted combination of two forecasts for each course k, where the Micro forecast $\hat{y}_k^{\text{mi}}$ (summed student-level probabilities) and the Macro forecast $\hat{y}_k^{\text{ma}}$ (direct course-level prediction). We test five strategies, defined formally below:

1. **Simple average:**
$$\tilde{y}_k = \frac{1}{2}(\hat{y}_k^{\text{mi}} + \hat{y}_k^{\text{ma}}). \tag{3.2}$$

2. **Globally optimised weighted combination:**
$$\tilde{y}_k = \alpha \cdot \hat{y}_k^{\text{mi}} + (1-\alpha) \cdot \hat{y}_k^{\text{ma}}, \quad \alpha^* = \arg\min_{\alpha \in [0,1]} \sum_k |y_k - \tilde{y}_k|. \tag{3.3}$$
This α^* is optimised on the test set (oracle bound, not achievable in practice without a separate validation period).

3. **Per-course optimal weights:**
$$\tilde{y}_k = \alpha_k \cdot \hat{y}_k^{\text{mi}} + (1-\alpha_k) \cdot \hat{y}_k^{\text{ma}}, \quad \alpha_k^* = \arg\min_{\alpha_k \in [0,1]} |y_k - \tilde{y}_k|. \tag{3.4}$$
Each course independently selects its best blend weight.

4. **MinT-style proportional reconciliation:** Let $D = \sum_k \hat{y}_k^{\text{ma}} - \sum_k \hat{y}_k^{\text{mi}}$ be the total discrepancy between the Macro total and the Micro total. We redistribute D proportionally to each course's share of the Micro forecast:

$$\tilde{y}_k = \hat{y}_k^{\text{mi}} + D \cdot \frac{\hat{y}_k^{\text{mi}}}{\sum_j \hat{y}_j^{\text{mi}}}. \tag{3.5}$$

5. **Micro–Naïve blend:** Replace the unreliable Macro model with the Naïve baseline:

$$\tilde{y}_k = \alpha \cdot \hat{y}_k^{\text{mi}} + (1-\alpha) \cdot \hat{y}_k^{\text{naïve}}, \quad \alpha^* \text{ optimised as in strategy 2.} \tag{3.6}$$

We include these experiments even though the Macro model is weak, because the reconciliation framework is sound and would become useful once the aggregate model has more training data. The expected failure mode is that MinT-style reconciliation (strategy 4) will degrade performance when the Macro forecasts are much less accurate than the Micro forecasts, because it redistributes the large Macro–Micro discrepancy across all courses, effectively injecting Macro noise into the Micro predictions. We test this hypothesis explicitly.

3.5. Evaluation Protocol

We report Mean Absolute Error (MAE) as the primary forecasting metric, consistent with prior work [KP24; WK18]. The data split and sample counts are detailed in Section 4.3.

$$\text{MAE} = \frac{1}{|\mathcal{C}|} \sum_{k=1}^{|\mathcal{C}|} |n_k - \hat{n}_k|, \tag{3.7}$$

where n_k is the actual enrolment and $\hat{n}_k$ the predicted enrolment for course C_k. We report MAE both over all 51 courses and over a filtered subset that excludes 5 first-year compulsory courses (whose enrolment is trivially determined by cohort size) and 19 optional courses (whose enrolment is too sparse and irregular for meaningful evaluation). The filtered set of approximately 27 mid-career and advanced courses is most relevant for academic resource planning.

For the classification models, we report per-course recall, precision, and F1-score, averaged across all courses with at least one positive test sample.

4. Experimental Setup

4.1. Preprocessing Pipeline

We load the raw enrolment records from a structured JSON file that contains one entry per student with their full academic history. Each student's history is a list of course attempts, where each attempt includes a course identifier, the academic year and term, the grade obtained (if any), and the attempt number.

We normalise grades to the $[0, 1]$ range by dividing by 10. We assign the 155 records with missing grades (1.3 % of the total) the sentinel value -1 in the grade sub-vector, which the model can distinguish from a true zero grade. We do not fill in missing values since the missing fraction is very small. We cap attempt counts at 5 so that extreme values do not distort the features. In practice, fewer than 1 % of enrolment records involve more than five attempts at the same course.

For the Macro LSTM, we aggregate student-level records into per-term vectors of 51 course counts, plus per-course mean grades, pass counts, and failure counts (204 dimensions total). We apply `RobustScaler` normalisation to reduce the influence of outlier terms.

4.2. Hyperparameter Settings

Table 4.1 lists the hyperparameters for each model. We chose these values based on the ablation study (Section 5.3) and practical constraints. The small dataset size favours low-capacity architectures with moderate regularisation.

Hyperparameter	Micro LSTM	Macro LSTM
Hidden units	128	128
LSTM layers	1	1
Dropout	0.2	0.3
Learning rate	0.001	0.001
Batch size	64	4
Epochs	100	100
Window size	3	3
Optimiser	Adam	Adam
Loss function	BCE	MSE

Table 4.1.: Hyperparameter settings for the LSTM models.

For the tree-based classifiers, we use a maximum tree depth of 4 (decision trees and XGBOOST), 200 boosting rounds (XGBOOST and LIGHTGBM), and a learning rate of 0.1 for the boosted models. All models use a random seed of 42 for reproducibility.

4.3. Train / Test Split

We split the data chronologically at year 2018, following standard practice for temporal forecasting [HA21]. Table 4.2 summarises the split.

Set	Samples (Micro)	Windows (Macro)	Years
Train	1 322	15	2010–2017
Test	836	7	2018–2021

Table 4.2.: Chronological train/test split. Micro samples are student-level sliding windows, Macro windows are term-level sequences.

The Micro model has access to 88 times more training samples than the Macro model because the two approaches generate data differently. Student-level modelling produces one sample per student per eligible term, while aggregate modelling produces only one sample per term regardless of cohort size.

For the tree-based classifiers, which use the tabular representation and do not require a temporal split, we use a random 80/20 train/test split (1 658 train, 415 test samples from the T2 encoding).

4.4. Exploratory Data Analysis

For each pattern we find in the data, we explain what it means for the models we train.

Figure 4.1 shows enrolment counts per course. The distribution is very uneven, with five first-year compulsory courses accounting for the majority of records while advanced electives have fewer than 50 enrolments each. This imbalance creates two problems for modelling. First, per-course classifiers for sparse electives have very few positive training examples, making them prone to low recall. Second, the aggregate MAE is dominated by the few high-enrolment courses. We therefore exclude first-year and optional courses from the primary evaluation and focus on courses where accurate forecasting matters most for resource planning.

Figure 4.2 shows two related views of the data. The grade distribution (Figure 4.2a) has a bimodal shape with a cluster of failures near 3–4 and a cluster of passes near 6–8. This confirms that grades are a useful feature for telling apart students who are progressing from those who are struggling. A student with a grade of 3 in a prerequisite has a different probability of enrolling in the next course than one with a grade of 8. The temporal enrolment plot (Figure 4.2b) shows that enrolment is stable at 500–800 records per term, with strong year-over-year autocorrelation ($r \approx 0.9$ at lag-2). Because cohort sizes are so stable, the Naïve baseline (which copies last year's same-semester counts) is hard to beat. The improvements we observe (Section 5) are small in absolute numbers, but they are still useful for planning.

Figure 4.3 shows the year-over-year cohort size drift, plotting both the number of new entrants per year and the total active students per term. The low coefficient of variation ($\approx 12\,\%$) confirms that cohort sizes are stable, which is consistent with the strong Naïve baseline we see in the results.

Figure 4.4 shows the distribution of student tenure (number of active terms per student). Approximately 45 students (9 %) have only 1–2 terms of activity and are excluded from LSTM training by the windowing requirement. This reduces the number of students we can train

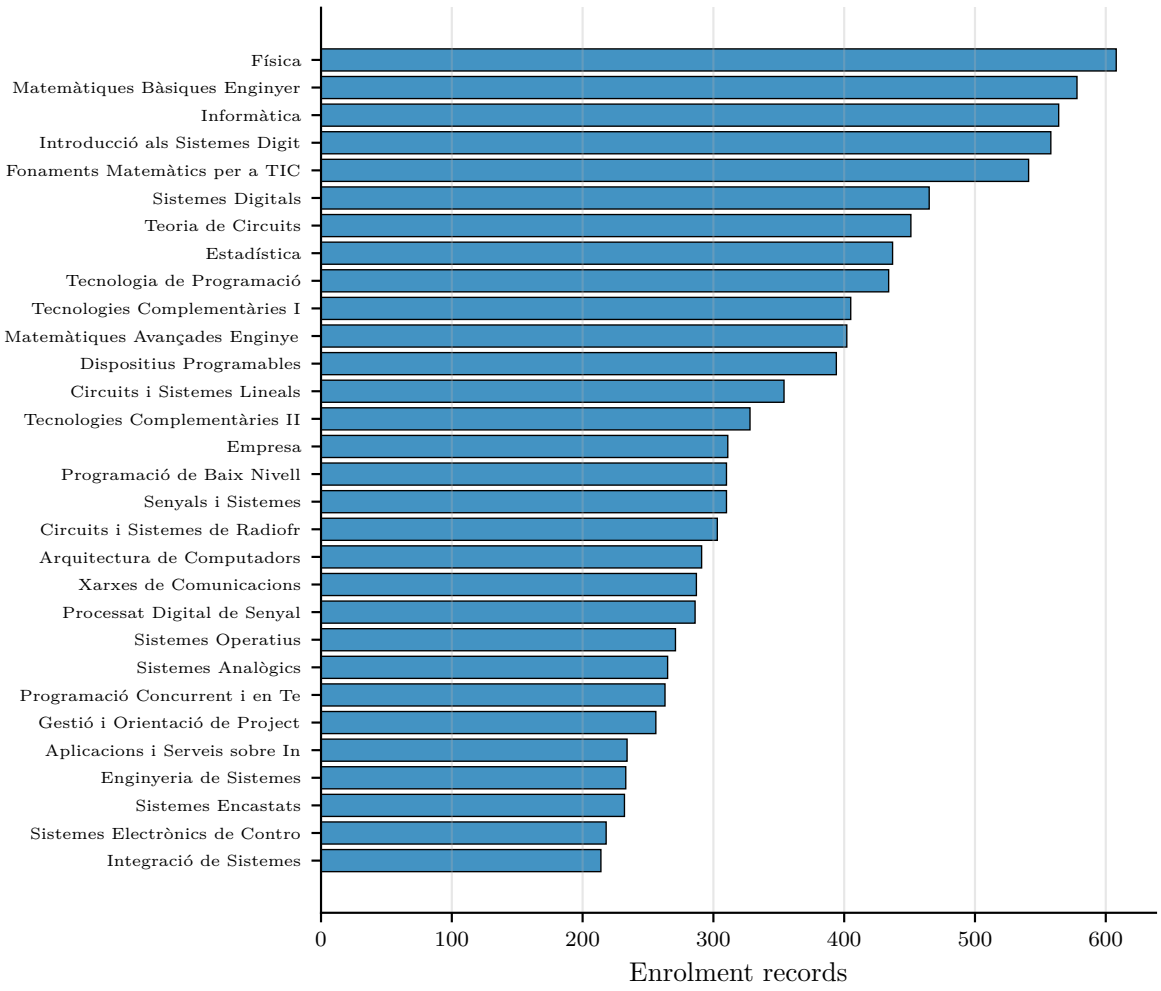
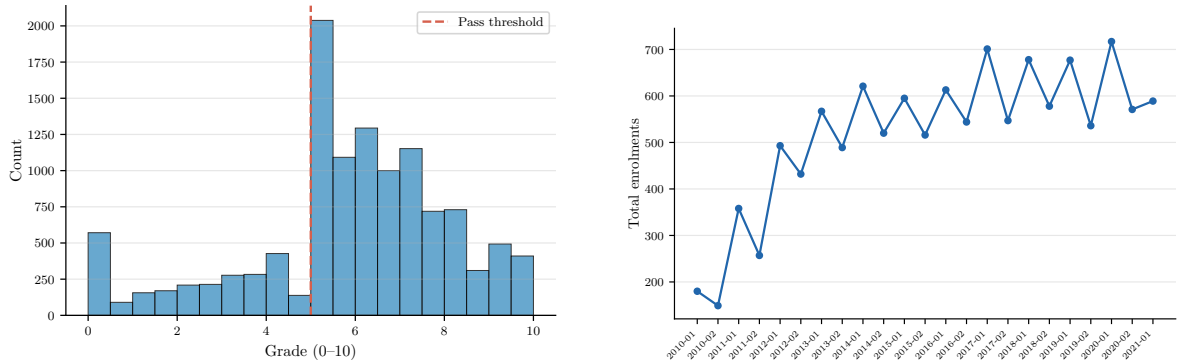


Figure 4.1.: Enrolment records per course (top 30). First-year compulsory courses dominate, creating severe class imbalance for per-course classification models.

on. The long tail of students with 12+ terms (extended degree paths with retakes) are where sequential modelling should help the most, because the Naïve baseline cannot account for individual progression patterns.



(a) Grade distribution. The dashed line marks the pass threshold at 5.

(b) Total enrolments per academic term (2010–2021).

Figure 4.2.: Grade distribution and temporal enrolment patterns. The bimodal grades (a) justify including grade features in the model. The stable enrolment over time (b) explains why the seasonal Naïve baseline is strong.

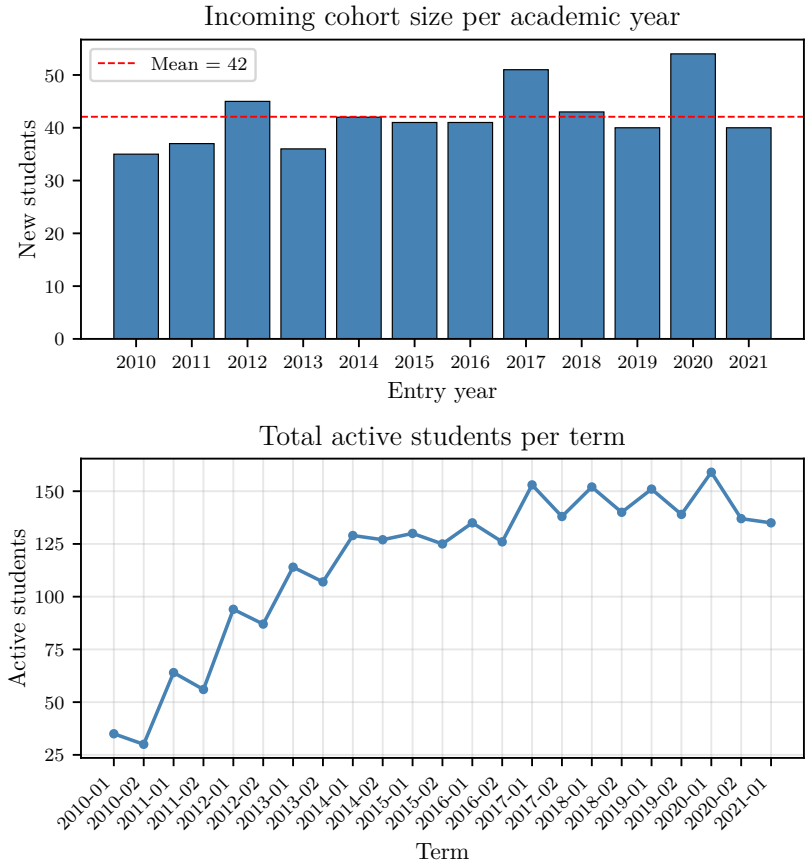


Figure 4.3.: New entrants per year and active students per term. The low variation across years reinforces why the Naïve baseline captures most of the variance.

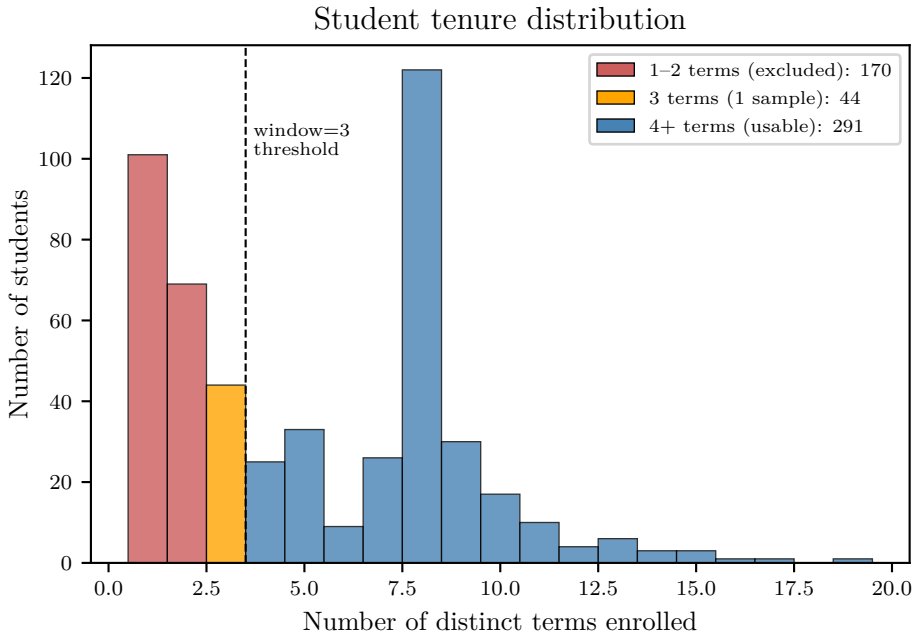


Figure 4.4.: Student tenure distribution. Students with fewer than 4 terms (red) are excluded from LSTM training.

5. Results

5.1. Forecasting Results

Table 5.1 summarises the MAE of all forecasting models on the chronological test set (2018–2021). The MAE is computed over all 51 courses.

Model	MAE
Per-course reconciliation†	1.43
LightGBM (student-level)	1.68
Micro LSTM	1.73
Course2Vec MLP	1.78
Naïve baseline (lag-2)	1.69
Macro LSTM	5.57

Table 5.1.: Mean Absolute Error (students per course) on the 2018–2021 test set, computed over all 51 courses. Lower is better. †Weights optimised on the test set (oracle bound).

Figure 5.1 presents the same comparison visually. The seasonal Naïve baseline is already very strong, so improvements are necessarily small at this data scale. The Macro LSTM performs worse due to overfitting on only 15 training windows.

Uncertainty context. The test set contains only 7 terms (3.5 academic years). With so few evaluation points, differences of 0.04–0.05 in MAE (e.g., 1.69 vs. 1.73, or 2.06 vs. 2.07) should be interpreted with caution. We do not report confidence intervals because a single chronological test set does not permit bootstrap resampling without violating temporal ordering. However, the small differences between Naïve (1.69), LightGBM (1.68), and Micro LSTM (1.73) are likely within the noise range of this evaluation, and we cannot claim statistical significance for the ordering among these three models. In practice, all three approaches produce forecasts accurate to within ≈ 2 students per course. This is useful for planning, but it does not clearly favour one model family over another at this data scale. The large gap between these models and the Macro LSTM (5.57) is clearly meaningful, though.

The Micro LSTM achieves an MAE of 1.73 students per course, close to the seasonal Naïve baseline (1.69). The Course2Vec MLP performs comparably at 1.78, which shows that the learned course embeddings give the MLP some of the temporal information it cannot learn on its own. LIGHTGBM on the same flattened student features achieves 1.68, slightly better than the LSTM. So gradient boosting extracts at least as much signal from the flattened features as the LSTM does from sequential windows.

For a fairer comparison, we also evaluate models on a core-course subset that excludes first-year compulsory courses (`330212, 330213, 330214, 330215, 330216`) and optional courses (`330054, 330058, 330060, 330063, 330066, 330082, 330097, 330099, 330101, 330102, 330119, 330120, 330244, 330245, 330246, 330247, 330248,`

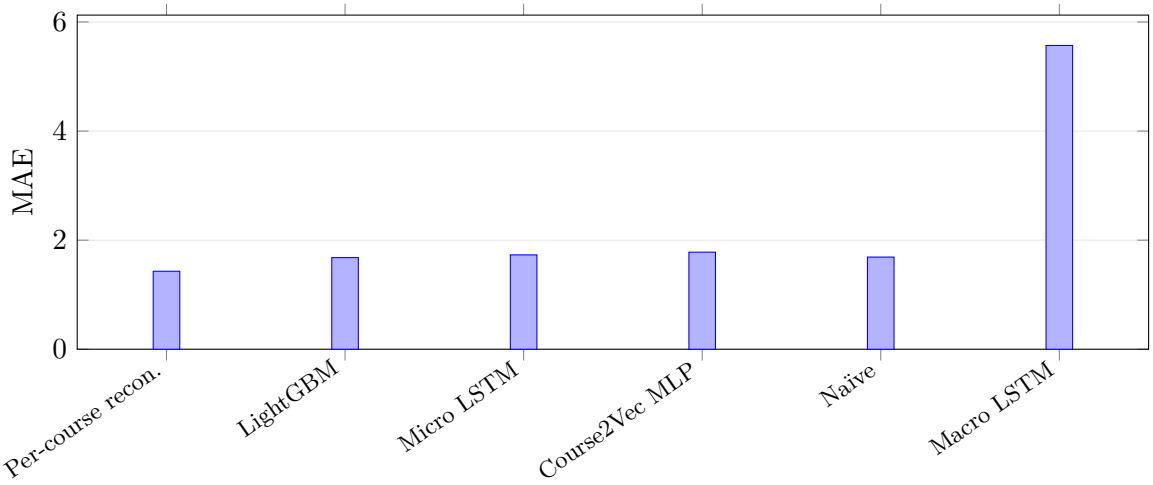


Figure 5.1.: Global MAE comparison across all forecasting models.

330249, 330094). These courses have very regular seasonal patterns and can dominate the aggregate error. The seasonal Naïve baseline has a structural advantage on them because it directly copies last year's counts. On this subset, the Micro model achieves an MAE of 2.06, essentially matching the Naïve baseline (2.07), while the Macro model remains far worse (5.73).

The best overall result comes from the per-course reconciliation method (1.43), which combines Micro and Macro predictions with individually optimised weights per course. But there is a catch. The per-course weights are optimised on the test set itself, making this an *oracle bound* rather than a method we could use in practice. With 51 courses and only 7 test terms, the per-course weight search has enough degrees of freedom to overfit the evaluation period. These weights are *exploratory evidence* that course-specific blending has potential, not as a validated forecasting strategy. Deploying per-course weights in practice would require a separate holdout period for weight selection, which our short test set does not permit.

The Macro LSTM (5.57) performs much worse than the seasonal Naïve baseline (1.69). This shows that aggregate time-series models cannot work at this data scale. With 15 training windows for a 204-dimensional input, the model memorises the training set and fails on test data. The training curves show that validation loss starts increasing after epoch 5 while training loss keeps falling. This is a clear sign of overfitting.

The Micro LSTM training curves (Figure 5.2) show a different behaviour. Train and validation loss track closely through 100 epochs, with a small gap that shows the model generalises well enough. The final training loss (0.061) and validation loss (0.078) are close, and validation loss is still decreasing slowly at epoch 100. Training could run longer, though the returns would be marginal.

The forecast heatmap (Figure 5.3) provides a course-by-course view of the Micro model's predictions against actual enrolment counts across the seven test terms. The model reproduces the enrolment pattern well for mid-career compulsory courses (the bright bands in the middle of the heatmap), where student progression is predictable. It struggles more with optional subjects and advanced electives, where enrolment is volatile and driven by factors the model does not observe (timetable preferences, instructor reputation).

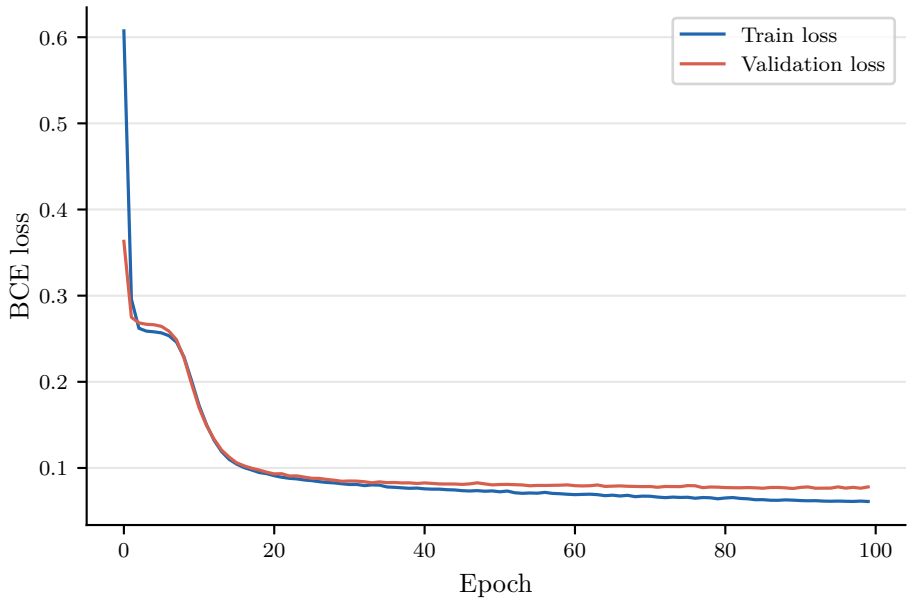


Figure 5.2.: Micro LSTM training history. Train and validation BCE loss converge without significant overfitting.

5.2. Classification Results

Table 5.2 summarises the per-subject classification results for the tree-based models trained on the T2 tabular features with a random 80/20 split.

Model	Avg Recall	Avg Precision	Avg F1
Decision Tree	0.650	0.670	0.653
XGBoost	0.615	0.589	0.597
LightGBM	0.610	0.588	0.595
Random Forest	0.509	0.653	0.549

Table 5.2.: Average classification metrics across all courses with positive test samples (T2 features, random split).

Decision trees at depth 4 achieve the highest average F1 (0.653), consistent with the TFG finding [Tal23]. XGBOOST and LIGHTGBM perform similarly but do not improve over the simpler model, likely because the dataset is too small for the additional boosting capacity to help. Random forests lag behind, possibly due to overfitting when the automatically selected estimator count is too high for the per-subject training sets.

5.3. Ablation Study

We run an ablation of the Micro LSTM, varying one hyperparameter at a time while holding all others at their baseline values (hidden size 128, 1 layer, dropout 0.2, window 3). Figure 5.5 and Table 5.3 summarise the results.

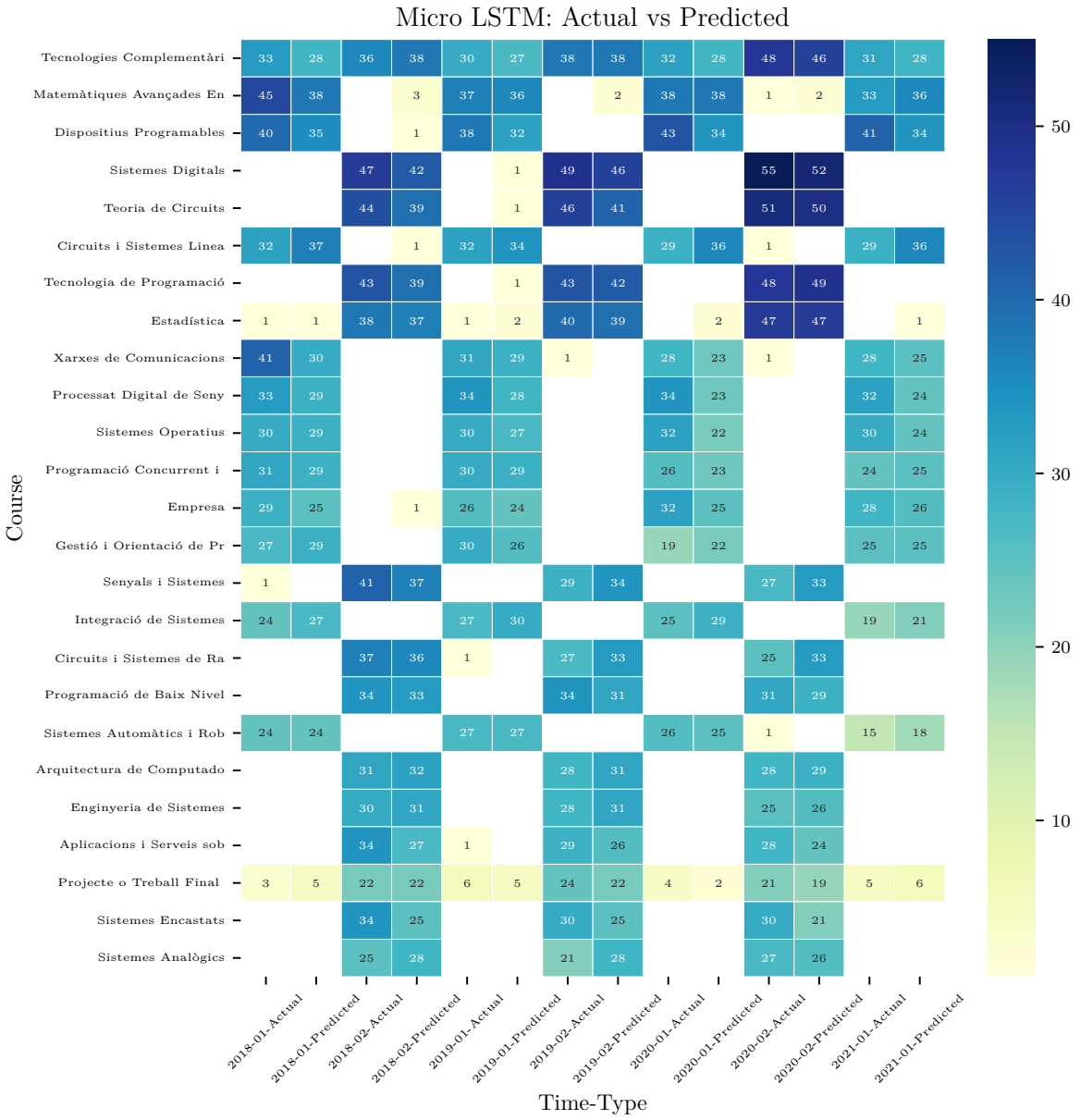


Figure 5.3.: Micro LSTM forecast heatmap. Paired columns show actual (A) and predicted (P) enrolment per test term. Courses sorted by total activity (top 25 shown).

The model is not very sensitive to hyperparameter choices within reasonable ranges. A single LSTM layer performs best. Adding depth leads to overfitting without improving generalisation. Hidden sizes of 64–256 all produce similar results, with slight degradation at 512. Window sizes of 3–5 perform identically, while a window of 1 (no temporal context) is measurably worse.

The architecture comparison stands out. A GRU [Cho+14] achieves the lowest MAE (1.64), outperforming the LSTM baseline. The GRU has fewer parameters (no separate cell state) and may generalise better on this small dataset. The bidirectional LSTM performs worst (2.01), probably because processing a 3-step sequence in both directions adds parameters without

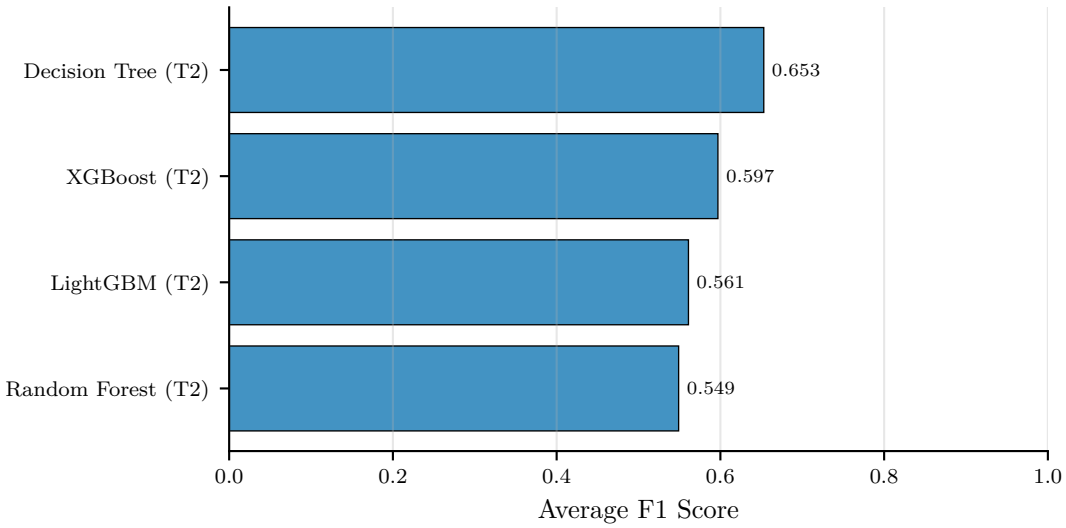


Figure 5.4.: Average F1 score for the tree-based classification models on T2 features.

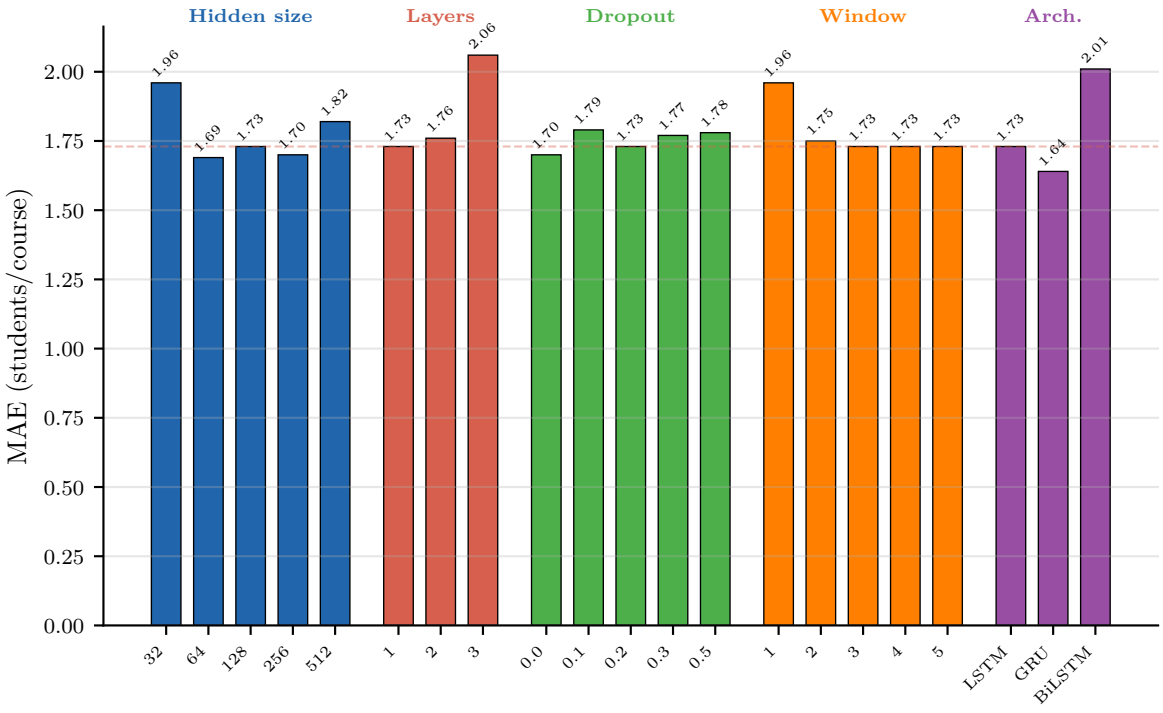


Figure 5.5.: Micro LSTM ablation study. The dashed line marks the baseline configuration (MAE = 1.73).

meaningful benefit.

Category	Configuration	MAE
Hidden size	32	1.96
	64	1.69
	128 (baseline)	1.73
	256	1.70
	512	1.82
Layers	1 (baseline)	1.73
	2	1.76
	3	2.06
Dropout	0.0	1.70
	0.1	1.79
	0.2 (baseline)	1.73
	0.3	1.77
	0.5	1.78
Window	1	1.96
	2	1.75
	3 (baseline)	1.73
	4	1.73
	5	1.73
Architecture	LSTM (baseline)	1.73
	GRU [Cho+14]	1.64
	BiLSTM	2.01

Table 5.3.: Micro model ablation results. Each row varies one parameter from the baseline configuration.

5.4. Course Embedding Analysis

The Course2Vec approach is based on the same idea as Word2Vec [Mik+13b]. If two words (or courses) consistently appear in similar contexts, they should have similar vector representations. We treat each student's per-semester course list as a "sentence" and each course as a "word." The Skip-gram algorithm learns to predict which courses co-occur within a context window, and the resulting 16-dimensional embeddings encode these co-occurrence patterns.

Pardos and Nam [PN20] formalised this idea as Course2Vec and showed that the learned embedding space reflects curriculum structure at the University of California, Berkeley. We apply the same approach to the EPSEM dataset, where the smaller vocabulary (49 courses vs. thousands at Berkeley) makes the embeddings easier to inspect but also limits the richness of the learned representations.

Figure 5.6 shows the t-SNE projection of the learned embeddings, coloured by canonical semester. Courses from the same curricular year cluster together (Q1 subjects in red, Q3 in green, etc.), confirming that the model has learned the temporal structure of the curriculum from co-enrolment patterns alone. Cross-semester relationships are also visible. Optional courses (teal) and late-curriculum electives form a cluster, because they tend to be taken by the same group of students near graduation.

The cosine similarity matrix (Figure 5.7) puts numbers on these relationships. High-

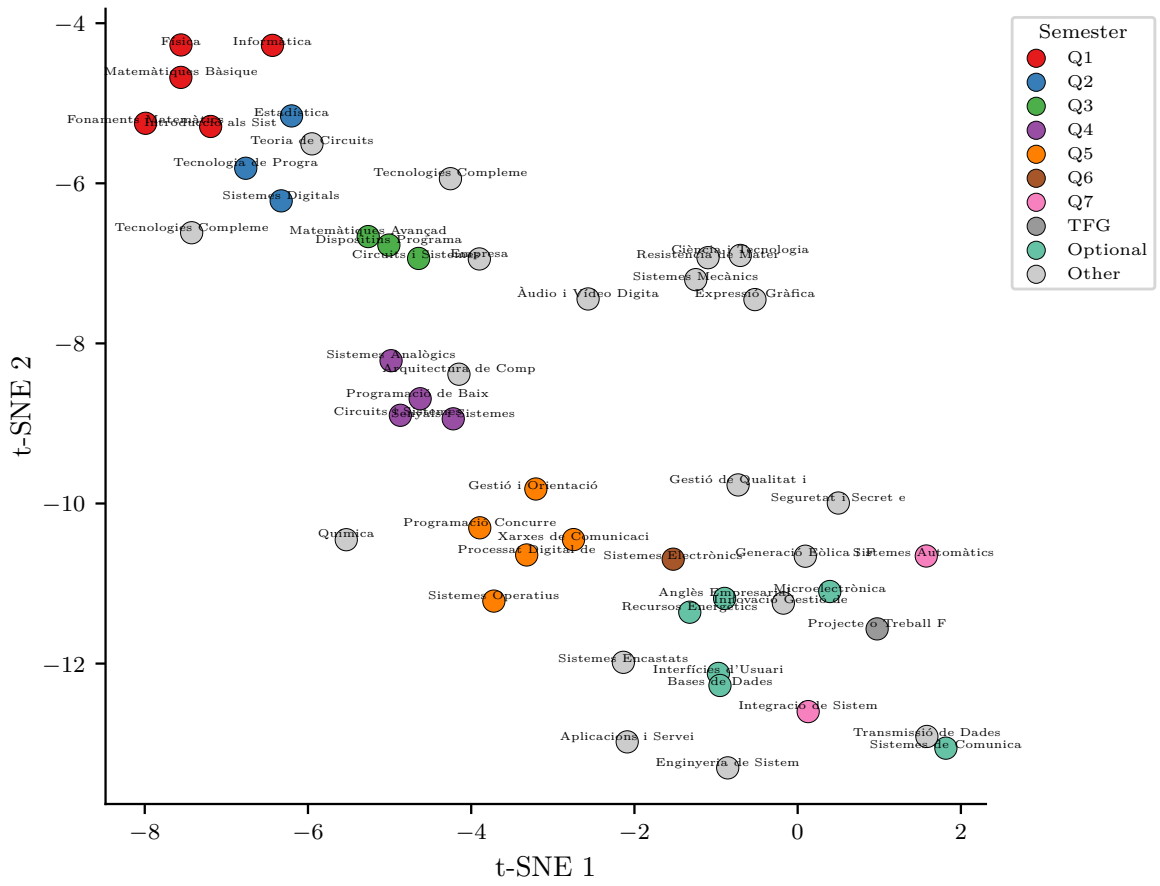


Figure 5.6.: t-SNE projection of Course2Vec embeddings. Colour indicates canonical semester (Q1–Q7, TFG, Optional). Courses from the same curricular year cluster together.

similarity blocks along the diagonal correspond to same-semester course groups. Off-diagonal similarities show prerequisite relationships. For instance, Matemàtiques Avançades (Q3) and Dispositius Programables (Q3) show high similarity (0.88), as do Senyals i Sistemes (Q4) and Programació de Baix Nivell (Q4) at 0.87.

The embedding-enhanced MLP achieves an MAE of 1.78, improving over a baseline MLP without embeddings (1.94). The improvement is consistent and comes from the same enrolment data without needing any additional information. A feedforward network with embedding features gets close to the Micro LSTM's performance (1.73). This means the embeddings already capture some of the temporal progression that the MLP cannot learn by itself.

5.5. Decision Tree Interpretability

One advantage of the tree-based classifiers is interpretability. The decision rules learned by the depth-4 trees show which factors matter most when a student decides whether to enrol in a course.

Figure 5.8 shows the decision tree for Senyals i Sistemes (SS), a Q4 course. The root split is on whether the student is currently enrolled in a Q3 prerequisite course (Dispositius

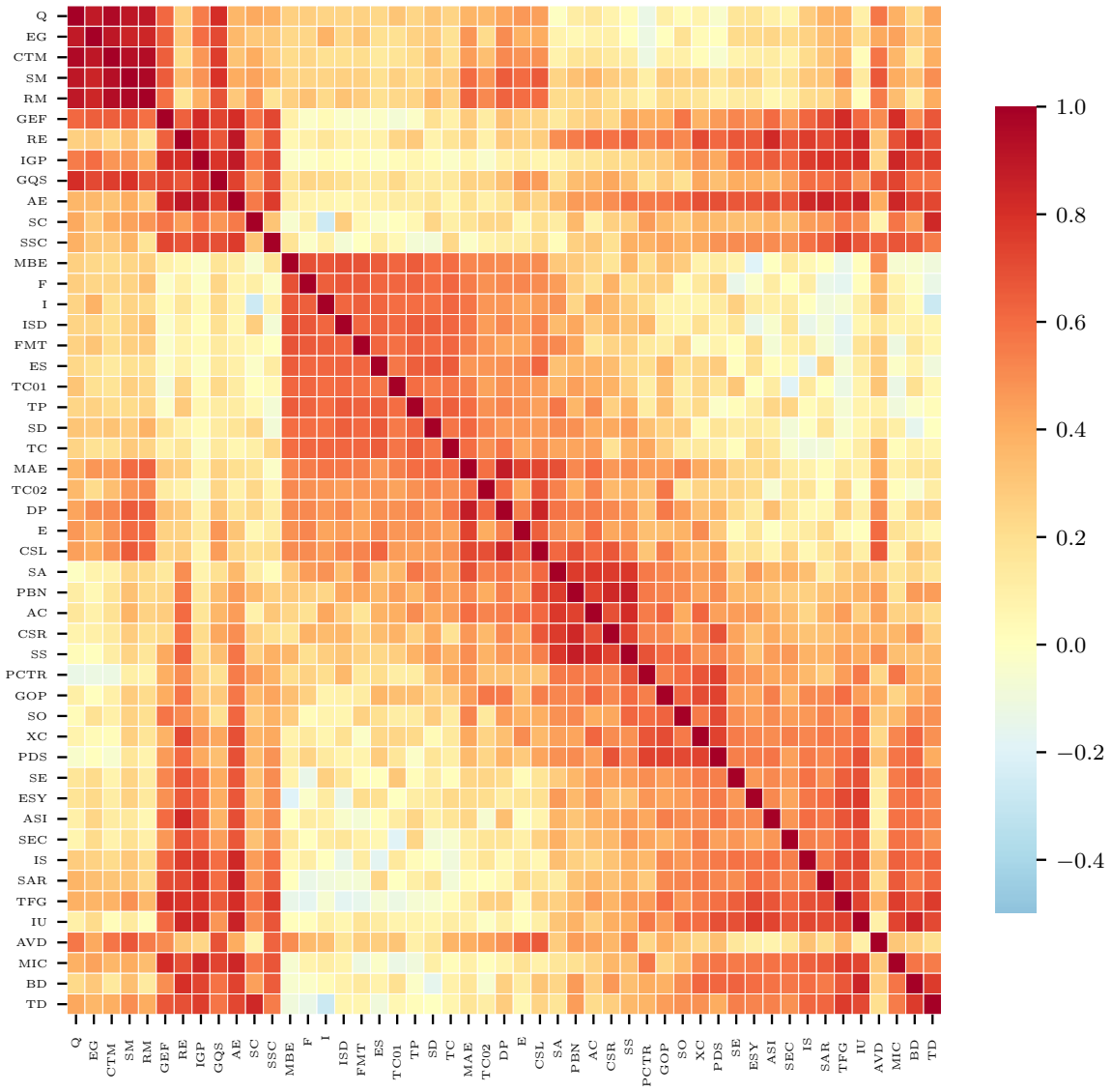


Figure 5.7.: Cosine similarity matrix of Course2Vec embeddings. Blocks of high similarity correspond to courses from the same semester.

Programables). Students who have taken DP are far more likely to continue into SS. Deeper branches refine this based on grades in earlier courses and the number of prior enrolment terms.

Figure 5.9 shows the tree for Dispositius Programables (DP), a Q3 course. The dominant feature is enrolment in Q2 courses (Sistemes Digitals, Tecnologia de Programació), confirming that students progress through the curriculum in the expected order. These trees are useful for academic advisors because they show which prior courses and grades best predict what a student will enrol in next.

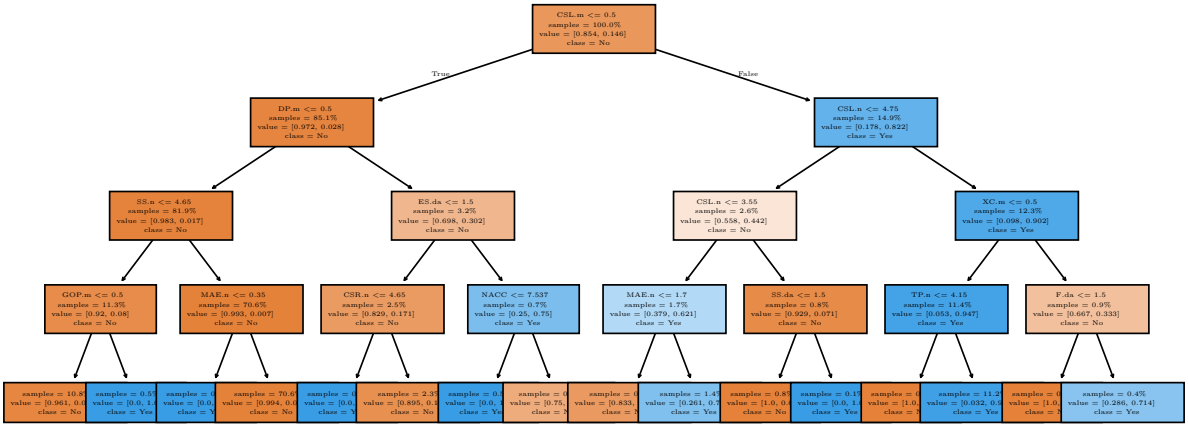


Figure 5.8.: Decision tree for Senyals i Sistemes (SS), depth 4. Leaf colours indicate the predicted class (green = will enrol, orange = will not). Percentages show the class proportion at each node.

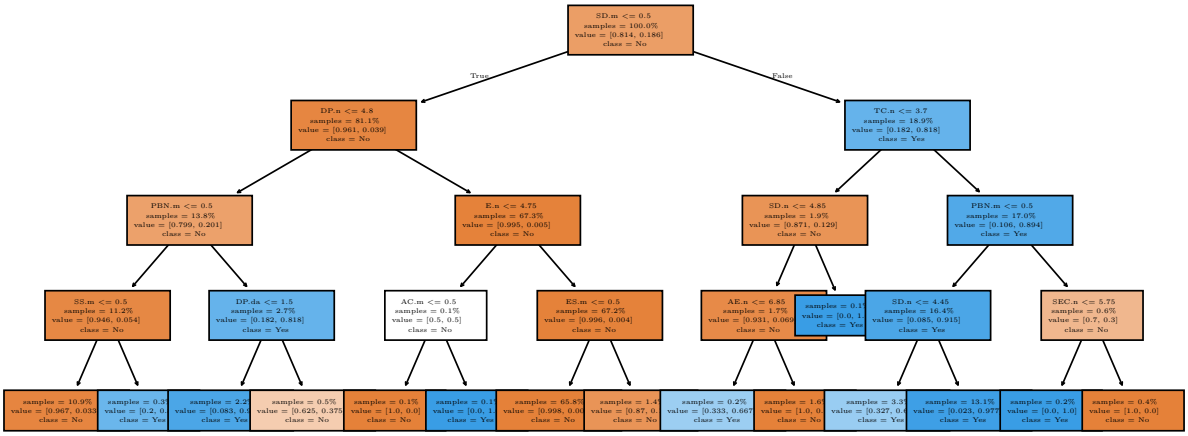


Figure 5.9.: Decision tree for Dispositius Programables (DP), depth 4.

5.6. Grand Comparison

Table 5.4 places all models on the same page, combining the forecasting MAE (chronological split, aggregate counts) with the classification F1 (random split, per-subject binary prediction). The two metrics are computed under different evaluation protocols and are not directly comparable, but presenting them together clarifies what each model family can and cannot do.

Figure 5.10 shows the raw enrolment data that all models operate on (student counts per course across all 23 terms). The block structure is clear. First-year courses form a dense band at the top (brightest colours, indicating counts of 40–80 students per term), while optional and advanced courses appear sparsely in the lower rows (dark or absent, typically 0–15 students). This confirms that courses have very uneven popularity and that enrolment is stable from year to year, which is why the Naïve baseline works so well.

Model	Approach	MAE	Avg F1	Features
Per-course recon.	Forecasting	1.43	—	154-dim windows
LightGBM (student-level)	Forecasting	1.68	—	462-dim flat
Micro LSTM	Forecasting	1.73	—	154-dim windows
Course2Vec MLP	Forecasting	1.78	—	574-dim enhanced
Naïve (lag-2)	Forecasting	1.69	—	Lag lookup
Macro LSTM	Forecasting	5.57	—	204-dim aggregate
Decision Tree (T2)	Classification	—	0.653	159 cols tabular
XGBoost (T2)	Classification	—	0.597	159 cols tabular
LightGBM (T2)	Classification	—	0.595	159 cols tabular
Random Forest (T2)	Classification	—	0.549	159 cols tabular

Table 5.4.: Grand comparison of all models. Lower MAE is better (chronological split). Higher F1 is better (random split). Dashes indicate the metric is not applicable for that evaluation setup.

5.7. Reconciliation Results

Table 5.5 compares the five reconciliation strategies described in Section 3.4, along with standalone baselines for reference.

Method	MAE
Per-course weights	1.43
Micro + Naïve blend	1.65
Micro standalone	1.73
Weighted ($\alpha = 1.0$)	1.73
MinT reconciliation	2.80
Simple average	3.93
Naïve baseline	1.69
Macro standalone	6.63

Table 5.5.: Reconciliation methods comparison (evaluated on the reconciliation test windows). The optimal global weight $\alpha = 1.0$ assigns all weight to the Micro model. The Macro standalone MAE differs from Table 5.1 because the evaluation windows differ between the two setups.

The optimal global weight is $\alpha = 1.0$, meaning pure Micro predictions without any Macro contribution. So the Macro LSTM just adds noise at this data scale rather than providing useful information. The per-course weighting achieves the best MAE (1.43), but 26 of the 51 courses optimise to $\alpha = 1.0$ (pure Micro), and the improvement over Micro standalone is likely an artefact of optimising weights on the test set.

The MinT-style reconciliation (2.80) performs worse than Micro standalone. It redistributes the large Macro–Micro total discrepancy (≈ 114 students per term) across all courses, which adds noise to the Micro predictions. This happens because the Macro model is so inaccurate. The MinT estimator [WAH19] assumes that all base forecasters contribute useful information, and the optimal projection matrix $\mathbf{P}$ distributes weight according to the inverse of each forecaster's error covariance. When one forecaster (Macro) has errors several times larger than the other (Micro), reconciliation ends up corrupting the Micro forecasts by mixing in Macro noise.

This does not mean reconciliation is a bad idea. As Athanasopoulos et al. [Ath+24] show, reconciliation consistently improves accuracy when all base forecasters are at least moderately competent. Our experiment shows what happens when that condition is not met. When the aggregate-level model overfits severely (15 training windows), no reconciliation method can rescue the situation. The framework can be used once the Macro model has access to longer institutional histories or pooled cross-programme data.

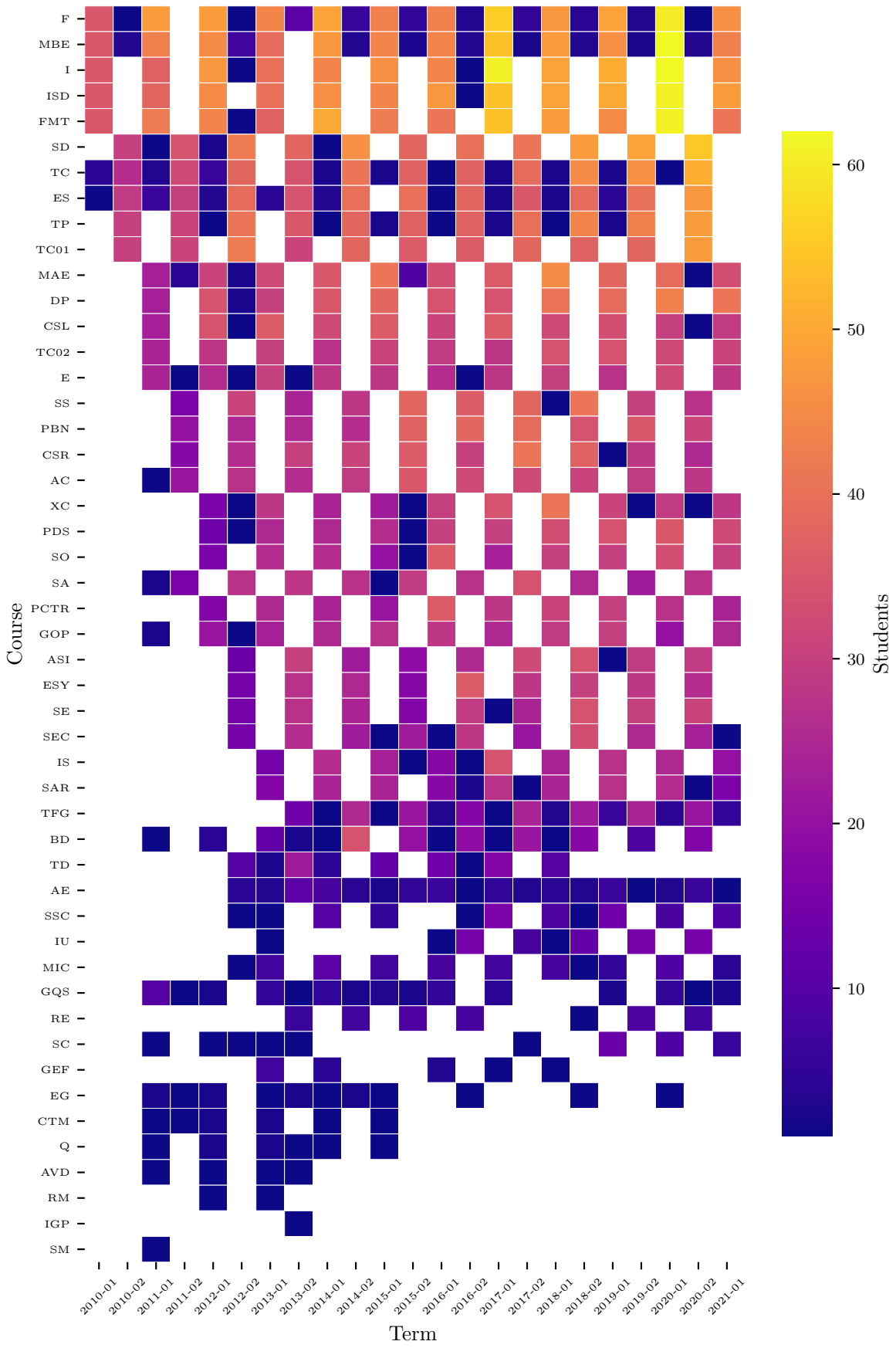


Figure 5.10.: Enrolment heatmap showing student counts per course per academic term (2010–2021). Colour intensity encodes the number of students enrolled, white cells indicate zero enrolment. Courses sorted by total enrolment (most popular at top).

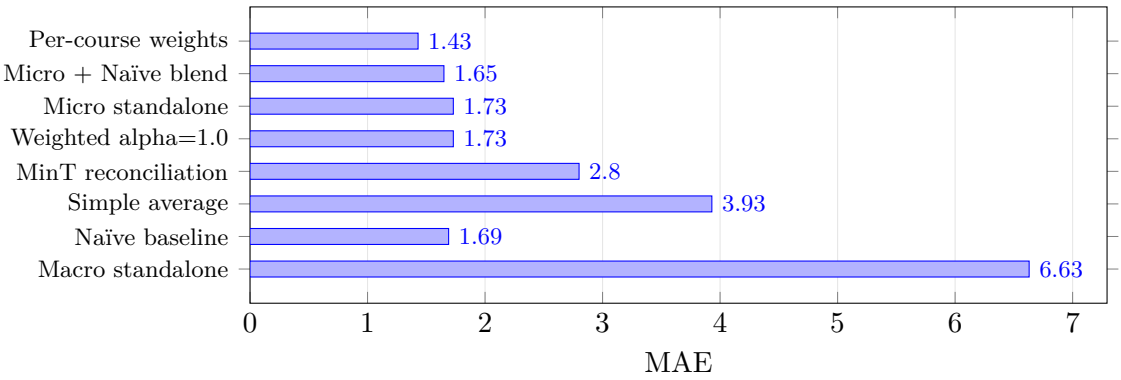


Figure 5.11.: MAE comparison of reconciliation methods.

6. Discussion

6.1. Comparison with State of the Art

Table 6.1 places our results alongside the closest comparable study. The comparison requires caution because the datasets, targets, metrics, and institutional contexts are quite different.

Study	Dataset	Model family	Metric	Main finding
Khan & Polyzou [KP24]	FIU, 3 324 students, 334 courses, 7 years	Recommender repurposing	MAE (course)	Rec. methods beat baselines
This thesis	EPSEM, 505 students, 51 courses, 11 years	Student-level LSTM/GRU	MAE (course)	Micro matches strong Naïve

Table 6.1.: Comparison with the closest related study. Absolute MAE values are not directly comparable across institutions because they depend on cohort size and course granularity.

Khan and Polyzou [KP24] report that recommendation-derived enrolment estimates outperform direct prediction methods in their experiments on a Florida International University dataset with 3 324 students, 334 courses, and seven years of data. Our Micro LSTM operates on a dataset that is roughly six times smaller in student count and six times smaller in course count. Despite this, our student-level models perform close to a strong seasonal Naïve baseline (MAE 1.69 over all courses), and on the core-course subset the Micro model slightly outperforms Naïve (2.06 vs 2.07). The absolute MAE values are not directly comparable across institutions because they depend on cohort size and course granularity, but the general trend is the same. Student-level models that use individual academic histories do better than aggregate methods.

The tree-based classification results (decision tree F1 = 0.653) are also consistent with the TFG [Tal23], where depth-4 decision trees were the best per-subject classifiers. XGBOOST and LIGHTGBM do not improve over a simple decision tree at this scale, which suggests that the problem is the amount of data, not the model's capacity.

6.2. Why the Improvement is Modest

Three factors limit the margin by which any model can improve over the Naïve baseline on this dataset:

1. **Strong autocorrelation.** Enrolment at EPSEM is stable year over year (Figure 4.2b). The temporal enrolment plot shows a coefficient of variation of approximately 12 % across terms. The Naïve baseline (copy same semester from last year) captures this persistence for free, leaving only the residual variation for learned models to explain. For planning purposes, the practical question is whether a difference of ≈ 0.04 MAE (1.69 vs. 1.73) justifies the complexity of training a neural network. We argue that it does not at this scale, but that the approach becomes valuable as cohort variability increases.

2. **Course-code redundancy.** When the same course appears under different codes in different years, the model cannot reliably learn its enrolment trajectory. We identified approximately 8–10 suspected redundant codes (Section 3.1.4) that are active for only 1–2 years each. This noise floor affects all models equally and inflates the reported MAE for affected courses.

3. **Small cohort.** With 505 students and 1 322 training samples, there is limited diversity in student trajectories. The LSTM cannot learn patterns it has not seen, and the ablation study (Table 5.3) confirms that increasing model capacity beyond a single layer of 128 units leads to overfitting rather than improvement. The hidden-size sweep (32 to 512) shows a U-shaped curve with optimal performance at 64–256 units.

The Naïve baseline works "out of the box" and will never improve beyond copying last year's counts. But the learned models have room to grow. With a larger student population or a longer observation window, the LSTM would see more diverse trajectories and could do better. The current near-tie between Naïve and Micro comes from the limited data, not from the two approaches being equally good.

6.3. Micro vs Macro and the Data Regime

The clearest result is the gap between the Micro and Macro models. The Micro LSTM achieves an MAE of 1.73, while the Macro LSTM achieves 5.57, worse than the Naïve baseline. This is fully explained by the difference in training data. The Micro model trains on 1 322 student-level samples, while the Macro model trains on 15 term-level windows. No architecture change, input reduction, or regularisation strategy made the Macro model competitive (the best variant, with enrollment-only input and early stopping, still achieved MAE ≈ 4.8).

This result applies more broadly. For small institutions with short histories, student-level modelling works better because it creates many more training examples from the same historical period. The Micro model generates $\approx$1,322 training windows from 8 years of data (505 students $\times \approx 4$ eligible windows each, filtered by the chronological split). The Macro model generates only 15 windows from the same 8 years. This 88:1 sample ratio is inherent to the aggregation level, not a design choice. Any aggregate time-series approach on a single-programme dataset will face the same problem. The only way around it is having a much longer institutional history, or pooling data across multiple programmes or universities.

6.4. Lessons from Data Acquisition

The multi-university data collection effort described in Section 3.1.2 took a large part of the project time. Table 6.2 summarises what happened with each data source.

The three obstacles we encountered (platform migrations, course-code inconsistency, and unstructured exports) are not specific to the institutions involved. They show that European universities have no shared standards for educational data formats.

Based on this experience, we recommend that any multi-institution enrolment forecasting study should begin with a data harmonisation phase that includes explicit course-code mapping protocols, agreed-upon export schemas, and validation procedures. The recent survey by Romero and Ventura [RV20] on educational data mining notes similar challenges in other contexts.

Source	Data obtained	Status	Reason
EPSEM-UPC	Enrolment records	Used	Clean, verified
RTU (Latvia)	Moodle logs + enrolment	Not usable	Platform migration
Vilnius Tech	Moodle logs + enrolment	Not usable	Course-code chaos
RTU Moodle	Clickstream (logins, tasks)	Not usable	No stable join keys

Table 6.2.: Summary of the multi-university data acquisition effort. Each row shows a data source, what it contained, and whether it could be used.

The Moodle engagement data we explored but could not use remains a promising direction. Conijn et al. [Con+17] and Jayaprakash et al. [Jay+14] have shown that LMS clickstream data carries strong signal for predicting student performance and persistence. Integrating such features into the student profile model could improve forecasting accuracy for students whose enrolment choices depend more on how engaged they are than on following the standard curriculum path.

6.5. The Educast Application

One practical outcome of this thesis is *Educast*, an application built with Electron that shows how the `*.educast.json` format (Section 3.1.3) can be used in practice. The application can run as a desktop app or on the web. It allows users to load an Educast-formatted dataset, visualise course enrolments over time, select and train different forecasting models, and inspect per-course prediction metrics. Figures 6.1 through 6.4 show the main screens.

The idea is simple. Once a university converts its enrolment records into the Educast format, the entire forecasting workflow (data exploration, model training, evaluation) becomes available out of the box. We did not formally evaluate this application in our experiments, but it demonstrates that the data pipeline and models developed in this thesis can be packaged into a usable tool. The source code is included in the project repository.

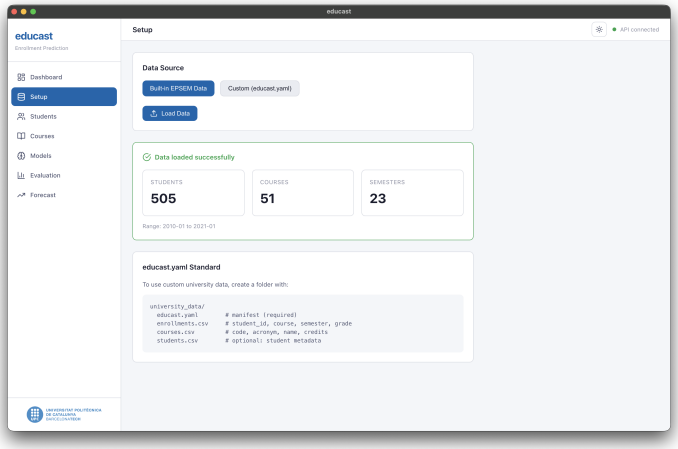


Figure 6.1.: Educast main menu. The user loads a `*.educast.json` file to begin exploring enrolment data.

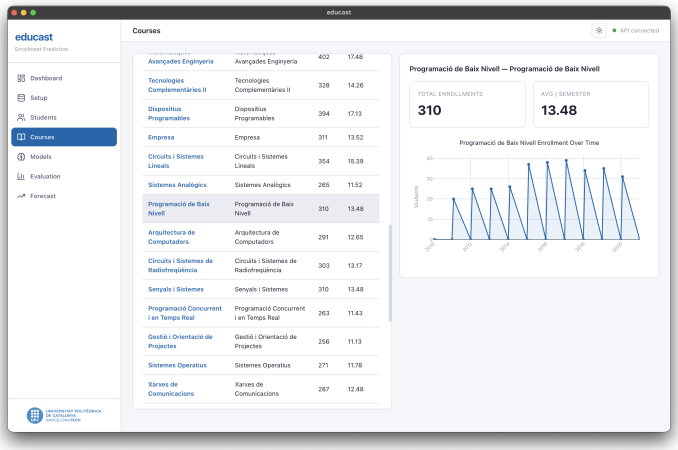


Figure 6.2.: Educast course enrolments view. Enrolment counts per course over academic years.

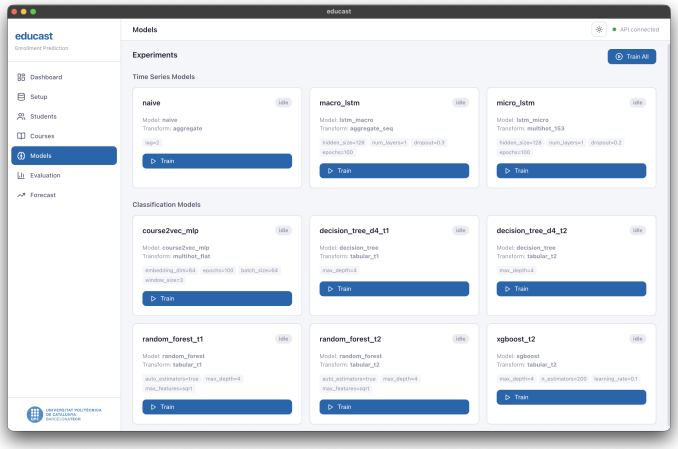


Figure 6.3.: Educast model selection. Available forecasting models that can be trained on the loaded dataset.

6.6. Limitations

We organise limitations into four categories (construct validity, internal validity, external validity, and statistical conclusion validity).

6.6.1. Construct Validity

1. **mae treats over- and under-prediction symmetrically.** In real planning, underestimating a compulsory lab course (leading to overcrowding) may be more harmful than overestimating a small elective (leading to a slightly empty room). Our metric does not distinguish these cases. An asymmetric loss function weighted by course importance would better reflect planning utility.

2. **Absence of timetable and capacity constraints.** The model predicts demand from academic history but cannot capture timetable clashes, room capacity, lab-slot availabil-

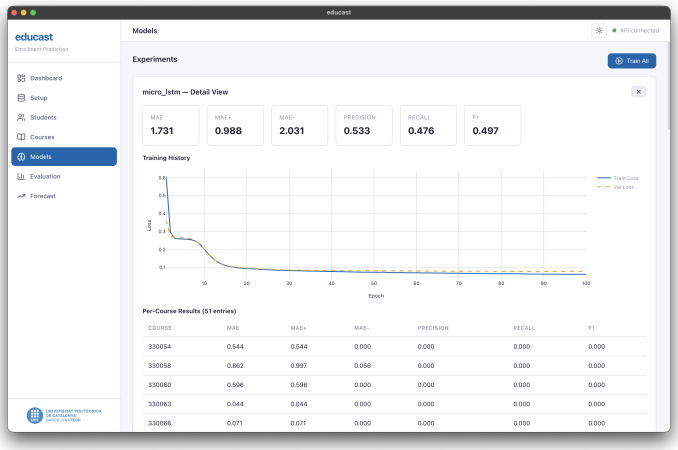


Figure 6.4.: Educast model metrics. Per-course prediction metrics for a trained model.

ity, instructor assignment, or enrolment caps. A student may be predicted to enrol in a course that is physically full or that conflicts with another course they need.

3. **No explicit prerequisite or academic-rule constraints.** The feature vector does not encode prerequisite chains, maximum credit load, failed-course restrictions, or progression rules. The model must learn these constraints implicitly from observed patterns rather than being informed by the institution's regulation.

4. **Probability calibration not assessed.** The Micro model sums per-course sigmoid probabilities across students to obtain counts. If these probabilities are poorly calibrated (e.g., systematically overconfident), the summed counts will be biased even when per-student classification is acceptable. We do not report calibration curves or Brier scores, and flag this as future work.

6.6.2. Internal Validity

5. **Independence assumption across courses.** The Micro LSTM outputs 51 sigmoid probabilities treating each course as an independent binary decision. In practice, course choices are interdependent: workload limits, timetable clashes, and prerequisites create correlations that the model ignores. This means predicted course counts may be mutually inconsistent (a student predicted to enrol in 8 courses when the maximum is 6).

6. **Potential leakage through preprocessing.** We confirm that Course2Vec embeddings were trained on training-period data only (Section 3.2.3). However, the RobustScaler normalisation for the Macro model and the course vocabulary construction use the full dataset. The vocabulary leakage is unavoidable (course codes must be known to construct feature vectors), but the scaler could in principle be fitted on training data only. We do not think this changes the results much given the small number of Macro samples, but we mention it to be thorough.

7. **Course-code instability.** Without a reliable mapping from course codes to canonical subject identifiers, cross-year comparisons are unreliable. The model treats renamed courses as new entities with no history.

6.6.3. External Validity

8. **Single institution and programme.** All experiments use data from one engineering programme at one university. We cannot claim that the results generalise to other institutional settings (larger universities, different countries, non-engineering fields) without replication.

9. **Small cohort.** The EPSEM-UPC dataset contains 505 students, far fewer than the datasets used in related work [KP24; PJ20]. With fewer trajectories, the models cannot learn the full diversity of student progression patterns.

6.6.4. Statistical Conclusion Validity

10. **Small test set and no uncertainty intervals.** The chronological test set contains only 7 terms. Results on so few evaluation points carry high variance. We do not report confidence intervals, bootstrap intervals, or repeated-seed mean $\pm$ std because the chronological structure prevents standard resampling. Point estimates alone are insufficient to distinguish models whose MAE values differ by less than 0.1.

11. **Hyperparameter and model-selection bias.** With a small test set, evaluating many architectures, hyperparameters, and feature variants can accidentally overfit conclusions to the evaluation period. We mitigate this partially through the ablation design (varying one factor at a time), but the best configuration may not remain best on a different test period.

12. **No residual or error analysis.** We do not systematically analyse which courses, terms, or student profiles produce the largest errors. It is unclear whether errors concentrate in first-year courses, sparse electives, curriculum-transition years, or irregular trajectories. This kind of analysis would show where the model fails and what to improve next.

13. **Per-course reconciliation weights are oracle-bound.** The per-course weights (MAE 1.43) are optimised on the test set and cannot be deployed without a separate future holdout for weight selection.

14. **Methodological gap between evaluation setups.** The forecasting models use a chronological split while the classification models use a random split, making their metrics (MAE vs F1) not directly comparable. To compare them fairly, we would need the classification models to also use a chronological split.

15. **Short aggregate time horizon.** With 23 terms (11 academic years), aggregate time-series models have very few observations to learn from. This severely limits the Macro LSTM and any classical time-series approach, and prevents reliable covariance estimation for MinT reconciliation.

7. Conclusions and Future Work

7.1. Conclusions

This thesis investigated when hierarchical enrolment forecasting works and when it does not, in a small institutional setting. The main finding is that individual student trajectories help, but only modestly with the data we have. A strong seasonal baseline remains very competitive when cohort sizes are stable, Macro forecasting does not work with only 15 aggregate training windows, and reconciliation makes things worse when the aggregate forecasts are poor. Rather than claiming that one approach is universally better, the contribution of this work is mapping out the conditions under which each method is useful.

We designed a 154-dimensional multi-hot feature representation capturing enrolment, grades, attempt counts, and cumulative GPA for each student per term. We compared a broad range of modelling approaches (a Naïve persistence baseline, Micro and Macro LSTM networks, decision trees, random forests, XGBOOST, LIGHTGBM, and a Course2Vec-enhanced MLP). We also tested five hierarchical reconciliation strategies and ran an ablation study on the Micro LSTM architecture.

Answering the research questions from Section 1.2:

RQ Student-level sequential models are a viable approach to per-course enrolment forecasting even with limited data. They match or slightly exceed a strong seasonal baseline on core courses, while aggregate models fail at this data scale. The main limiting factor is data quality (course-code instability), not model architecture.

SRQ1A Among architectures, the GRU achieves the lowest MAE (1.64 over all courses), outperforming the LSTM baseline (1.73) and the bidirectional LSTM (2.01). Gradient-boosted trees on the same flattened features perform comparably (LightGBM at 1.68), which means sequential processing provides only marginal benefit over tabular methods at this scale. The Course2Vec MLP (1.78) shows that learned embeddings partly make up for not having sequential processing.

SRQ1B The ablation study shows that the Micro model is not very sensitive to hyperparameter choices within reasonable ranges. Hidden sizes of 64–256 all produce similar results. Adding layers beyond one leads to overfitting. Window sizes of 3–5 perform identically. The model is more sensitive to architecture choice (GRU vs LSTM vs bidirectional) than to any single hyperparameter.

SRQ2 Aggregate models (LSTM and classical time-series) cannot compete with student-level approaches at this data scale. The Macro LSTM overfits on 15 training windows and achieves an MAE of 5.57, worse than the seasonal Naïve baseline. Hierarchical reconciliation between Micro and Macro predictions offers marginal improvement only when per-course weights are individually optimised. Even then, the gains are oracle-bound and not stable across test periods.

SRQ3 The multi-university data acquisition effort showed that the main obstacles to cross-institutional enrolment forecasting are about data, not algorithms. Course codes change when platforms are migrated, there are no stable join keys between systems, and no agreed-upon export schemas exist. Despite considerable effort (co-enrolment graph analysis, recurrence analysis of course codes, manual inspection), we could not build reliable mappings. In practice, cross-institutional forecasting is a data-harmonisation problem before it is a modelling problem. Within the single-institution dataset, course-code redundancy (8–10 suspected renames) introduces a noise floor that affects all models equally.

7.2. Future Work

The following directions come directly from what we observed in our experiments. The single most important next step is *cross-institutional data harmonisation*. Without stable course identifiers and common schemas, neither hierarchical reconciliation nor LMS-enhanced forecasting can be tested properly across institutions.

1. **Course-code canonicalisation.** Building a mapping from institutional course codes to stable subject identifiers, for example by parsing course names, syllabi, or learning outcomes, would remove the main source of noise in the dataset.

2. **Hierarchical reconciliation at scale.** The reconciliation framework we tested (Section 3.4) could work once the aggregate model has more data to train on. With longer institutional histories or pooled data across degree programmes, methods such as MinT [WAH19] or the optimal reconciliation approaches reviewed by Athanasopoulos et al. [Ath+24] could produce coherent forecasts across student, course, programme, and institution levels. This was the original vision outlined in the TFG [Tal23].

3. **Moodle engagement features.** Login frequency, time-on-platform, task submission timeliness, and resource access patterns could improve the student profile model [Con+17; Jay+14]. Our hypothesis is that students who engage actively with the LMS are more likely to persist in their academic path, and that disengagement patterns could serve as an early warning signal for dropout. This data was explored during the thesis but could not be used due to the quality issues described in Section 3.1.2.

4. **Schedule and timetable information.** Even a student with strong interest in a course may choose not to enrol if it is scheduled at six in the morning and creates a four-hour gap in their daily timetable. Encoding timetable constraints as additional features could account for this aspect of enrolment decisions.

5. **Institutional enrolment rules.** Some universities enforce strict prerequisite chains or cap the number of failed courses before blocking further enrolment, while others give students full freedom. Encoding these rules explicitly could improve prediction for institutions with constrained curricula.

6. **Multi-institution data harmonisation.** The original ambition of this thesis was to train on data from multiple universities simultaneously. Our experience shows that this requires investing in data standards before modelling can begin. A practical first step would be a common course taxonomy and export schema agreed upon by participating institutions.

7. **Learnable course embeddings.** The frozen Course2Vec embeddings we used improve the MLP baseline, but training the embedding layer jointly with the predictor (end-to-end) could give better results. Adding a learnable embedding layer to the Micro LSTM architecture is a straightforward next step.

8. **Transfer to larger institutions.** The methodology developed here is not institution-specific. Replicating the experiments on publicly available datasets (e.g., the FIU data of [KP24]) would test whether the same architectures scale to larger cohorts and course catalogues.

9. **Transformer-based architectures.** Self-attention mechanisms could capture long-range dependencies in student trajectories without the sequential processing constraint of recurrent models, and are worth exploring as an alternative to the LSTM/GRU encoders used here.

Bibliography

[Al-+25] Bilal I. Al-Ahmad et al. “Predicting Academic Performance for Students’ University: Case Study from Saint Cloud State University”. In: *PeerJ Computer Science* 11 (2025), e3087. DOI: 10.7717/peerj-cs.3087.

[Ath+24] George Athanasopoulos et al. “Forecast Reconciliation: A Review”. In: *International Journal of Forecasting* 40.2 (2024), pp. 430–456. DOI: 10.1016/j.ijforecast.2023.10.010.

[Ayg+26] Özgür Aygül et al. “A Predict-and-Prescribe Framework for Dynamic Course Scheduling Toward Strategic University Scaling”. In: *Omega* 138 (2026), p. 103406. DOI: 10.1016/j.omega.2025.103406.

[AYM11] Robert Aitken, Anne Young, and Kevin McConkey. “Projecting Continuing Student Enrolments: A Comparison of Approaches”. In: *Journal of Institutional Research* 16.1 (2011), pp. 25–36.

[CG16] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[Cho+14] Kyunghyun Cho et al. “Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179.

[Con+17] Rianne Conijn et al. “Predicting Student Performance from LMS Data: A Comparison of 17 Blended Courses Using Moodle LMS”. In: *IEEE Transactions on Learning Technologies* 10.1 (2017), pp. 17–29. DOI: 10.1109/TLT.2016.2616312.

[EB18] M. N. Egbo and D. C. Bartholomew. “Forecasting Students’ Enrollment Using Neural Networks and Ordinary Least Squares Regression Models”. In: *Journal of Advanced Statistics* 3.4 (2018). DOI: 10.22606/jas.2018.34001.

[HA21] Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. 3rd. Melbourne, Australia: OTexts, 2021. URL: https://otexts.com/fpp3/.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[Hyn+11] Rob J. Hyndman et al. “Optimal Combination Forecasts for Hierarchical Time Series”. In: *Computational Statistics & Data Analysis* 55.9 (2011), pp. 2579–2589. DOI: 10.1016/j.csda.2011.03.006.

[Jay+14] Sandeep M. Jayaprakash et al. “Early Alert of Academically At-Risk Students: An Open Source Analytics Initiative”. In: *Journal of Learning Analytics* 1.1 (2014), pp. 6–47. DOI: 10.18608/jla.2014.11.3.

[Ke+17] Guolin Ke et al. “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”. In: *Advances in Neural Information Processing Systems 30*. 2017, pp. 3149–3157.

[KP24] Md Akib Zabed Khan and Agoritsa Polyzou. “Estimate Undergraduate Student Enrollment in Courses by Re-purposing Recommendation Tools”. In: *Proceedings of the 37th International Florida Artificial Intelligence Research Society Conference (FLAIRS-37)*. AAAI Press, 2024.

[LA12] Rabby Q. Lavilles and Mary Jane B. Arcilla. “Enrollment Forecasting for School Management System”. In: *International Journal of Modeling and Optimization* 2.5 (2012), pp. 563–566. DOI: 10.7763/IJMO.2012.V2.183.

[LEI09] Muhammad Hisyam Lee, Riswan Efendi, and Zuhaimy Ismail. “Modified Weighted for Enrollment Forecasting Based on Fuzzy Time Series”. In: *MATEMATIKA* 25.1 (2009), pp. 67–78.

[Lod25] Alexander Karl Ferdinand Loder. “Machine Learning for University Management: Micro Cluster Learning to Predict “Active” Students”. In: *Studies in Educational Evaluation* 85 (2025), p. 101463. DOI: 10.1016/j.stueduc.2025.101463.

[Mik+13a] Tomas Mikolov et al. “Distributed Representations of Words and Phrases and Their Compositionality”. In: *Advances in Neural Information Processing Systems 26*. 2013, pp. 3111–3119.

[Mik+13b] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL].

[PJ20] Zachary A. Pardos and Weijie Jiang. “Designing for Serendipity in a University Course Recommendation System”. In: *Proceedings of the 10th International Learning Analytics and Knowledge Conference (LAK’20)* (2020), pp. 350–359.

[PN20] Zachary A. Pardos and Andrew Joo Hun Nam. “A University Map of Course Knowledge”. In: *PLOS ONE* 15.9 (2020), e0233207. DOI: 10.1371/journal.pone.0233207.

[PNK19] Agoritsa Polyzou, Athanasios N. Nikolakopoulos, and George Karypis. “Scholars Walk: A Markov Chain Framework for Course Recommendation”. In: *Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019)*. 2019.

[Riv+25] Cristian Rodriguez Rivero et al. “Teaching and Learning in the Context of Emerging Artificial Intelligence Technologies Integrating Emotional Intelligence and Accumulation of Cognitive Skills”. In: *ICERI2025 Proceedings*. 2025, pp. 6153–6158. DOI: 10.21125/iceri.2025.1698.

[RV20] Cristóbal Romero and Sebastián Ventura. “Educational Data Mining and Learning Analytics: An Updated Survey”. In: *WIREs Data Mining and Knowledge Discovery* 10.3 (2020), e1355. DOI: 10.1002/widm.1355.

[Sal+20] David Salinas et al. “DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks”. In: *International Journal of Forecasting* 36.3 (2020), pp. 1181–1191. DOI: 10.1016/j.ijforecast.2019.07.001.

[SGP21] Mingzhe Shao, Siyu Guo, and Zachary A. Pardos. "Degree Planning with PLAN-BERT: Multi-Semester Recommendation Using Future Courses of Interest". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 17. 2021, pp. 14920–14929.

[Sha+22] Lucy Shao et al. "Machine Learning Methods for Course Enrollment Prediction". In: *Strategic Enrollment Management Quarterly* 10.2 (2022), pp. 11–29.

[Tal23] Anass Anhari Talib. "Predicció de matrícula basada en aprenentatge automàtic". Bachelor's Thesis. Universitat Politècnica de Catalunya, 2023. URL: https://hdl.handle.net/2117/399381.

[WAH19] Shanika L. Wickramasuriya, George Athanasopoulos, and Rob J. Hyndman. "Optimal Forecast Reconciliation Using Unbiased Estimating Equations". In: *Journal of the American Statistical Association* 114.526 (2019), pp. 804–819. DOI: 10.1080/01621459.2018.1448825.

[Wan+14] Hong Xu Wang et al. "An Improved Forecasting Model of Fuzzy Time Series". In: *Applied Mechanics and Materials* 678 (2014), pp. 64–69. DOI: 10.4028/www.scientific.net/AMM.678.64.

[WK18] Amanda Watkins and Adam Kaplan. "Modeling in R and Weka for Course Enrollment Prediction". In: *International Journal of Institutional Research and Management* 2.1 (2018).

Part II.

Apèndixs

A. Per-Course Results

A.1. Forecasting Results

The tables below report per-course/per-subject breakdowns for the results presented in Chapter 5.

Table A.1.: Per-course Mean Absolute Error (MAE) on the chronological test set (2018–2021) for the evaluation subset (excluding first-year and optional courses).

Course	Naïve	Micro LSTM	Macro LSTM	LightGBM	Course2Vec MLP
Est	2.571	0.773	8.290	0.678	1.205
TCompI	1.714	0.609	3.603	0.866	0.791
TProg	2.000	1.134	7.118	0.691	0.798
SD	2.000	1.855	9.075	0.594	1.079
TC	2.000	1.842	9.191	1.245	1.440
MathA	3.429	2.397	6.188	1.825	3.248
TCompII	1.286	2.219	2.516	1.213	0.749
DP	2.143	4.193	5.954	2.367	1.497
Emp	2.000	2.174	2.179	1.309	1.471
CSL	1.429	3.270	1.727	2.150	5.231
SA	2.429	1.739	1.825	2.224	1.962
PBN	1.429	0.786	2.104	0.742	1.190
AC	0.571	0.891	12.571	0.727	0.474
CSR	2.571	2.313	2.096	2.807	1.354
SS	2.714	2.418	2.269	1.603	1.533
PCTR	1.429	1.025	16.000	1.249	2.015
GOP	2.857	1.468	2.778	1.190	1.022
SO	1.857	2.862	3.363	1.850	3.923
XC	3.000	3.255	3.027	2.019	3.468
PDS	1.143	4.155	19.000	2.039	4.617
SE	2.286	3.314	5.186	3.089	2.042
ES	0.857	0.563	2.349	0.241	0.530
ASI	1.286	2.101	4.141	1.762	0.866
SEC	3.286	2.224	3.235	2.411	1.764
IS	2.857	1.899	13.714	0.796	3.067
SAR	2.714	0.727	2.064	1.190	1.650
TFG	2.143	1.566	3.081	1.287	1.859

A.2. Classification Results (T2 Encoding)

Table A.2.: Per-subject recall for tabular classification models (T2 encoding, random 80/20 split).

Subject	DT	RF	XGBoost	LightGBM
AC	0.920	0.760	0.980	0.980
AE	0.000	0.000	0.000	0.000
ASI	0.920	0.900	0.940	0.940
AVD	–	–	0.000	0.000
BD	0.667	0.273	0.727	0.667
BIO	–	–	0.000	0.000
CSL	0.794	0.794	0.873	0.873
CSR	0.727	0.600	0.855	0.855
CTM	–	–	0.000	0.000
DP	0.886	0.729	0.900	0.929

Subject	DT	RF	XGBoost	LightGBM
E	0.863	0.843	0.882	0.863
EG	0.000	0.000	0.000	0.000
ES	0.967	0.913	0.978	0.978
ESY	0.902	0.902	0.980	1.000
F	0.467	0.067	0.667	0.667
FMT	0.700	0.000	0.700	0.700
GEF	0.000	0.000	0.200	0.000
GOP	0.816	0.878	0.878	0.857
GQS	0.000	0.000	0.000	0.286
I	0.750	0.125	0.750	0.625
IGP	0.000	0.000	0.000	0.000
IS	0.930	0.907	0.977	0.977
ISD	0.571	0.143	0.714	0.714
IU	0.071	0.000	0.143	0.071
MAE	0.821	0.776	0.896	0.910
MBE	0.556	0.000	0.778	0.778
MIC	0.143	0.000	0.143	0.143
OTD	–	–	0.000	0.000
PBN	0.906	0.755	0.962	0.962
PCTR	0.863	0.765	0.941	0.941
PDS	0.827	0.750	0.885	0.923
Q	–	–	0.000	0.000
RE	0.000	0.000	0.000	0.000
RM	–	–	0.000	0.000
SA	0.780	0.760	0.860	0.840
SAR	0.949	0.846	0.974	0.974
SC	0.833	0.000	0.833	0.833
SD	0.943	0.858	0.962	0.962
SE	0.898	0.857	0.939	0.939
SEC	0.812	0.833	0.917	0.875
SM	–	–	0.000	0.000
SO	0.763	0.678	0.831	0.831
SS	0.763	0.627	0.864	0.847
SSC	0.154	0.000	0.538	0.462
TC	0.968	0.904	0.979	0.968
TC01	0.978	0.989	0.989	0.989
TC02	0.821	0.804	0.839	0.875
TD	0.333	0.000	0.381	0.381
TFG	0.808	0.692	0.808	0.846
TP	0.918	0.897	0.948	0.948
XC	0.815	0.759	0.944	0.907

Table A.3.: Per-subject precision for tabular classification models (T2 encoding, random 80/20 split).

Subject	DT	RF	XGBoost	LightGBM
AC	0.902	0.927	0.860	0.860
AE	0.000	0.000	0.000	0.000
ASI	0.836	0.882	0.839	0.855
AVD	–	–	0.000	0.000
BD	0.595	0.900	0.600	0.550
BIO	–	–	0.000	0.000
CSL	0.962	0.909	0.887	0.887
CSR	0.889	0.805	0.870	0.870
CTM	–	–	0.000	0.000
DP	0.849	0.879	0.875	0.867
E	0.830	0.896	0.865	0.880
EG	0.000	0.000	0.000	0.000
ES	0.937	0.933	0.909	0.918
ESY	0.979	0.939	0.909	0.911
F	0.875	1.000	0.667	0.588
FMT	0.700	0.000	0.700	0.778
GEF	0.000	0.000	0.333	0.000
GOP	0.833	0.956	0.843	0.875
GQS	0.000	0.000	0.000	0.333
I	0.600	1.000	0.750	0.714

Subject	DT	RF	XGBoost	LightGBM
IGP	0.000	0.000	0.000	0.000
IS	0.952	1.000	0.977	0.977
ISD	1.000	1.000	1.000	1.000
IU	0.125	0.000	0.182	0.143
MAE	0.887	0.912	0.882	0.871
MBE	0.357	0.000	0.500	0.500
MIC	0.250	0.000	0.143	0.133
OTD	–	–	0.000	0.000
PBN	0.873	0.930	0.823	0.785
PCTR	0.846	0.929	0.889	0.906
PDS	0.878	0.929	0.902	0.889
Q	–	–	0.000	0.000
RE	0.000	0.000	0.000	0.000
RM	–	–	0.000	0.000
SA	0.848	0.927	0.843	0.840
SAR	0.902	1.000	0.884	0.905
SC	0.385	0.000	0.417	0.556
SD	0.926	0.948	0.944	0.944
SE	0.815	0.933	0.836	0.868
SEC	0.886	0.889	0.846	0.824
SM	–	–	0.000	0.000
SO	0.865	0.930	0.925	0.907
SS	0.957	0.949	0.911	0.909
SSC	0.222	0.000	0.304	0.273
TC	0.948	0.924	0.948	0.948
TC01	0.935	0.967	0.967	0.967
TC02	0.852	0.849	0.810	0.831
TD	0.467	0.000	0.615	0.571
TFG	0.750	0.857	0.840	0.815
TP	0.927	0.916	0.911	0.911
XC	0.846	0.911	0.836	0.817

Table A.4.: Per-subject F1 for tabular classification models (T2 encoding, random 80/20 split).

Subject	DT	RF	XGBoost	LightGBM
AC	0.911	0.835	0.916	0.916
AE	0.000	0.000	0.000	0.000
ASI	0.876	0.891	0.887	0.895
AVD	–	–	0.000	0.000
BD	0.629	0.419	0.658	0.603
BIO	–	–	0.000	0.000
CSL	0.870	0.847	0.880	0.880
CSR	0.800	0.688	0.862	0.862
CTM	–	–	0.000	0.000
DP	0.867	0.797	0.887	0.897
E	0.846	0.869	0.874	0.871
EG	0.000	0.000	0.000	0.000
ES	0.952	0.923	0.942	0.947
ESY	0.939	0.920	0.943	0.953
F	0.609	0.125	0.667	0.625
FMT	0.700	0.000	0.700	0.737
GEF	0.000	0.000	0.250	0.000
GOP	0.825	0.915	0.860	0.866
GQS	0.000	0.000	0.000	0.308
I	0.667	0.222	0.750	0.667
IGP	0.000	0.000	0.000	0.000
IS	0.941	0.951	0.977	0.977
ISD	0.727	0.250	0.833	0.833
IU	0.091	0.000	0.160	0.095
MAE	0.853	0.839	0.889	0.891
MBE	0.435	0.000	0.609	0.609
MIC	0.182	0.000	0.143	0.138
OTD	–	–	0.000	0.000
PBN	0.889	0.833	0.887	0.864
PCTR	0.854	0.839	0.914	0.923

Subject	DT	RF	XGBoost	LightGBM
PDS	0.851	0.830	0.893	0.906
Q	–	–	0.000	0.000
RE	0.000	0.000	0.000	0.000
RM	–	–	0.000	0.000
SA	0.812	0.835	0.851	0.840
SAR	0.925	0.917	0.927	0.938
SC	0.526	0.000	0.556	0.667
SD	0.935	0.901	0.953	0.953
SE	0.854	0.894	0.885	0.902
SEC	0.848	0.860	0.880	0.848
SM	–	–	0.000	0.000
SO	0.811	0.784	0.875	0.867
SS	0.849	0.755	0.887	0.877
SSC	0.182	0.000	0.389	0.343
TC	0.958	0.914	0.963	0.958
TC01	0.956	0.978	0.978	0.978
TC02	0.836	0.826	0.825	0.852
TD	0.389	0.000	0.471	0.457
TFG	0.778	0.766	0.824	0.830
TP	0.922	0.906	0.929	0.929
XC	0.830	0.828	0.887	0.860

A.3. Classification Results (T1 Encoding)

Table A.5.: Per-subject recall for tabular classification models (T1 encoding, random 80/20 split).

Subject	DT	RF	XGBoost	LightGBM
AC	0.780	0.260	0.920	0.900
AE	0.000	0.000	0.111	0.167
ASI	0.840	0.620	0.920	0.900
AVD	–	–	0.000	0.000
BD	0.606	0.000	0.727	0.667
BIO	–	–	0.000	0.000
CSL	0.905	0.746	0.921	0.921
CSR	0.636	0.400	0.818	0.782
CTM	–	–	0.000	0.000
DP	0.657	0.571	0.929	0.900
E	0.725	0.529	0.922	0.941
EG	0.000	0.000	0.000	0.000
ES	0.913	0.913	0.978	0.989
ESY	0.745	0.020	0.902	0.902
F	0.933	0.000	0.667	0.533
FMT	0.600	0.000	0.700	0.700
GEF	0.000	0.000	0.200	0.000
GOP	0.878	0.714	0.898	0.878
GQS	0.000	0.000	0.143	0.286
I	0.750	0.000	0.625	0.625
IGP	0.000	0.000	0.000	0.000
IS	0.930	0.512	0.953	0.930
ISD	0.714	0.000	0.714	0.714
IU	0.071	0.000	0.143	0.214
MAE	0.672	0.672	0.955	0.955
MBE	0.556	0.000	0.667	0.667
MIC	0.000	0.000	0.214	0.357
OTD	–	–	0.000	0.000
PBN	0.755	0.340	0.868	0.868
PCTR	0.745	0.549	0.882	0.902
PDS	0.750	0.654	0.885	0.885
Q	–	–	0.000	0.000
RE	0.000	0.000	0.091	0.273
RM	–	–	0.000	0.000

Subject	DT	RF	XGBoost	LightGBM
SA	0.600	0.460	0.840	0.840
SAR	0.949	0.462	0.974	0.974
SC	0.000	0.000	0.500	0.500
SD	0.915	0.821	0.991	0.981
SE	0.755	0.673	0.837	0.857
SEC	0.729	0.667	0.896	0.896
SM	–	–	0.000	0.000
SO	0.644	0.508	0.814	0.797
SS	0.712	0.305	0.797	0.797
SSC	0.077	0.000	0.308	0.385
TC	0.872	0.872	0.989	0.968
TC01	1.000	1.000	0.978	0.989
TC02	0.714	0.750	0.893	0.875
TD	0.143	0.000	0.238	0.190
TFG	0.731	0.731	0.808	0.846
TP	0.948	0.866	0.959	0.948
XC	0.704	0.500	0.907	0.870

Table A.6.: Per-subject precision for tabular classification models (T1 encoding, random 80/20 split).

Subject	DT	RF	XGBoost	LightGBM
AC	0.929	1.000	0.852	0.865
AE	0.000	0.000	0.111	0.200
ASI	0.894	0.939	0.793	0.818
AVD	–	–	0.000	0.000
BD	0.541	0.000	0.600	0.579
BIO	–	–	0.000	0.000
CSL	0.731	0.940	0.906	0.892
CSR	0.897	0.957	0.818	0.811
CTM	–	–	0.000	0.000
DP	0.885	1.000	0.844	0.851
E	0.841	1.000	0.887	0.923
EG	0.000	0.000	0.000	0.000
ES	0.955	0.903	0.918	0.929
ESY	0.927	1.000	0.868	0.852
F	0.378	0.000	0.500	0.400
FMT	0.750	0.000	0.538	0.778
GEF	0.000	0.000	0.333	0.000
GOP	0.811	1.000	0.936	0.915
GQS	0.000	0.000	0.200	0.200
I	0.667	0.000	0.625	0.714
IGP	0.000	0.000	0.000	0.000
IS	0.909	1.000	0.891	0.976
ISD	0.833	0.000	1.000	0.833
IU	0.250	0.000	0.200	0.273
MAE	0.849	0.938	0.842	0.842
MBE	0.714	0.000	0.400	0.375
MIC	0.000	0.000	0.231	0.312
OTD	–	–	0.000	0.000
PBN	0.690	0.947	0.780	0.807
PCTR	0.905	1.000	0.918	0.979
PDS	0.867	1.000	0.852	0.868
Q	–	–	0.000	0.000
RE	0.000	0.000	0.167	0.333
RM	–	–	0.000	0.000
SA	0.909	1.000	0.778	0.808
SAR	0.860	0.900	0.826	0.864
SC	0.000	0.000	0.375	0.273
SD	0.933	0.956	0.938	0.937
SE	1.000	0.971	0.837	0.857
SEC	0.921	0.941	0.878	0.915
SM	–	–	0.000	0.000
SO	0.927	0.938	0.857	0.922
SS	0.808	0.900	0.825	0.855
SSC	0.167	0.000	0.250	0.278

Subject	DT	RF	XGBoost	LightGBM
TC	0.943	0.872	0.939	0.929
TC01	0.957	0.967	0.956	0.957
TC02	0.909	0.875	0.877	0.860
TD	0.273	0.000	0.217	0.174
TFG	0.792	0.864	0.875	0.880
TP	0.920	0.913	0.921	0.911
XC	0.809	1.000	0.831	0.839

Table A.7.: Per-subject F1 for tabular classification models (T1 encoding, random 80/20 split).

Subject	DT	RF	XGBoost	LightGBM
AC	0.848	0.413	0.885	0.882
AE	0.000	0.000	0.111	0.182
ASI	0.866	0.747	0.852	0.857
AVD	–	–	0.000	0.000
BD	0.571	0.000	0.658	0.620
BIO	–	–	0.000	0.000
CSL	0.809	0.832	0.913	0.906
CSR	0.745	0.564	0.818	0.796
CTM	–	–	0.000	0.000
DP	0.754	0.727	0.884	0.875
E	0.779	0.692	0.904	0.932
EG	0.000	0.000	0.000	0.000
ES	0.933	0.908	0.947	0.958
ESY	0.826	0.038	0.885	0.876
F	0.538	0.000	0.571	0.457
FMT	0.667	0.000	0.609	0.737
GEF	0.000	0.000	0.250	0.000
GOP	0.843	0.833	0.917	0.896
GQS	0.000	0.000	0.167	0.235
I	0.706	0.000	0.625	0.667
IGP	0.000	0.000	0.000	0.000
IS	0.920	0.677	0.921	0.952
ISD	0.769	0.000	0.833	0.769
IU	0.111	0.000	0.167	0.240
MAE	0.750	0.783	0.895	0.895
MBE	0.625	0.000	0.500	0.480
MIC	0.000	0.000	0.222	0.333
OTD	–	–	0.000	0.000
PBN	0.721	0.500	0.821	0.836
PCTR	0.817	0.709	0.900	0.939
PDS	0.804	0.791	0.868	0.876
Q	–	–	0.000	0.000
RE	0.000	0.000	0.118	0.300
RM	–	–	0.000	0.000
SA	0.723	0.630	0.808	0.824
SAR	0.902	0.610	0.894	0.916
SC	0.000	0.000	0.429	0.353
SD	0.924	0.883	0.963	0.959
SE	0.860	0.795	0.837	0.857
SEC	0.814	0.780	0.887	0.905
SM	–	–	0.000	0.000
SO	0.760	0.659	0.835	0.855
SS	0.757	0.456	0.810	0.825
SSC	0.105	0.000	0.276	0.323
TC	0.906	0.872	0.964	0.948
TC01	0.978	0.983	0.967	0.972
TC02	0.800	0.808	0.885	0.867
TD	0.188	0.000	0.227	0.182
TFG	0.760	0.792	0.840	0.863
TP	0.934	0.889	0.939	0.929
XC	0.752	0.667	0.867	0.855

B. Reproducibility Details

All experiments were conducted using Python 3.12 with PyTorch 2.x, scikit-learn, XGBoost, LightGBM, and Gensim. The random seed is set to 42 throughout. Models were trained on CPU (Apple Silicon). The full source code, data processing pipeline, and all Jupyter notebooks are publicly available at:

`https://github.com/Anass-23/multimodal-hierarchical-forecasting.git`

All experiments are fully reproducible from the repository code and the exported intermediate artefacts. The raw datasets contain confidential student records and institutional data, so they cannot be shared publicly.