USING LEARNING ANALYTICS & MACHINE LEARNING TO ENHANCE EARLY DETECTION OF AT-RISK STUDENTS IN HIGHER EDUCATIONAL INSTITUTIONS

Tu Lam

Thesis submitted to the University of Ottawa in partial
fulfillment of the requirements for the degree of
Master of Science (MSc) in Digital Transformation and Innovation

Faculty of Engineering
University of Ottawa

Abstract

The rapid transition to remote learning during the Coronavirus Disease of 2019 (COVID-19) pandemic significantly disrupted student-instructor interaction, contributing to increased dropout rates in higher education. According to the 2022 Canadian Student Wellbeing Study, 72% of students experienced reduced in-person engagement, and 40% considered withdrawing due to insufficient institutional support. In response, Learning Analytics (LA) has gained prominence as a data-driven approach to enhancing student engagement, academic performance, and retention. This thesis addresses two core challenges: the need for a scalable LA framework and the development of an effective Machine Learning (ML)-based LA to identify at-risk students.

Despite growing interest in LA, many studies overlook foundational implementation challenges and the limited capabilities of existing LA tools, such as Brightspace's Student Success System (S3). To address these gaps, this study begins with a review of LA adoption in global and Canadian contexts, highlighting that while over half of Canadian Higher Education Institutions (HEI) are engaging in LA initiatives, empirical evidence of impact remains scarce. The thesis proposes a comprehensive LA architecture tailored to The University of Ottwawa (uOttawa), emphasizing early risk detection and addressing functional gaps in current Learning Management System (LMS) tools.

Building on this foundation, the core of this work involves the development of ML model for at-risk student prediction. Following the Cross-Industry Standard Process for Data Mining (CRISP-DM), six classification algorithms are evaluated across four temporal checkpoints in the academic term. The models incorporate features from the Student Information System (SIS) and Brightspace LMS, including demographic attributes, academic history, and time-aware performance metrics. Feature importance analysis reveals a dynamic shift from reliance on historical SIS data to LMS-derived grade indicators as the course progresses. Notably, time-based features—especially phase-specific grade averages—proved crucial for late-phase predictions. Among the models tested, Random Forest (RF) and Extreme Gradient Boosting (XGB) consistently achieved the highest recall and accuracy, identifying 75–91% of at-risk students with overall accuracy between 60–81%.

This thesis offers three key contributions: (1) a critical review of LA adoption and implementation strategies; (2) a proposed institutional LA architecture to support predictive analytics; (3) an empirical methodology of comparing classification models and time-aware feature dynamics, as a foundation for a deployable Early Warning System (EWS) to inform proactive interventions for improving student success and retention.

Résumé

La transition rapide vers l'apprentissage à distance durant la pandémie de COVID-19 a profondément perturbé l'interaction entre étudiants et enseignants, contribuant à une hausse des taux d'abandon dans l'enseignement supérieur. Selon l'étude canadienne sur le bien-être des étudiants de 2022, 72 % des étudiants ont constaté une diminution de l'engagement en présentiel, et 40 % ont envisagé d'abandonner leurs études en raison d'un manque de soutien institutionnel. En réponse, l'analyse de l'apprentissage (LA) s'est imposée comme une approche fondée sur les données pour améliorer l'engagement, la performance académique et la rétention des étudiants. Cette thèse aborde deux défis majeurs : le besoin d'un cadre LA évolutif et le développement d'un système efficace basé sur l'apprentissage automatique (ML) pour identifier les étudiants à risque.

Bien que l'intérêt pour l'analyse de l'apprentissage ne cesse de croître, de nombreuses études négligent les défis liés à l'implémentation initiale et les limitations fonctionnelles des outils existants comme S3 dans Brightspace. Pour combler ces lacunes, ce travail commence par une revue de l'adoption de LA dans les contextes internationaux et canadiens, révélant que plus de la moitié des établissements postsecondaires canadiens mènent des initiatives LA, mais que les preuves empiriques de leur impact restent limitées. La thèse propose une architecture complète d'analyse de l'apprentissage adaptée à l'Université d'Ottawa, mettant l'accent sur la détection précoce des risques et la résolution des insuffisances fonctionnelles des outils actuels de gestion de l'apprentissage (LMS).

Sur cette base, le cœur de cette recherche porte sur le développement d'un modèle ML pour la prédiction des étudiants à risque. En suivant le processus CRISP-DM, six algorithmes de classification sont évalués à quatre moments clés du trimestre académique. Les modèles exploitent des caractéristiques provenant du SIS et du LMS Brightspace, incluant les données démographiques, les antécédents académiques et des indicateurs de performance temporels. L'analyse d'importance des variables révèle une transition dynamique de la dépendance aux données historiques du SIS vers des indicateurs de notes dérivés du LMS au fil du cours. Les caractéristiques temporelles, notamment les moyennes spécifiques à chaque phase, se sont révélées essentielles pour les prédictions tardives. Parmi les modèles évalués, la forêt aléatoire (RF) et XGBoost (XGB) ont obtenu les meilleurs résultats en termes de rappel et de précision, identifiant entre 75 % et 91 % des étudiants à risque avec une précision globale comprise entre 60 % et 81 %.

Cette thèse présente trois contributions principales : (1) une revue critique des stratégies d'adoption et d'implémentation de l'analyse de l'apprentissage ; (2) une architecture institutionnelle LA proposée pour soutenir l'analyse prédictive ; (3) une méthodologie empirique

pour comparer les modèles de classification et la dynamique temporelle des variables, en vue de la mise en œuvre d'un système d'alerte précoce (EWS) favorisant des interventions proactives pour améliorer la réussite et la rétention des étudiants.

Statement of Contributors of Collaborators and Co - Authorship

I declare that I am the sole author of this thesis. I gratefully acknowledge the supervision of Prof. Andrew Sowinski, whose guidance was instrumental throughout the development of this work. His support included editorial feedback on each chapter, as well as valuable input in shaping the research objectives and overall direction of the thesis.

Acknowledgements

I would like to express my heartfelt gratitude to my supervisor, Professor Andrew Sowinski, for his continuous guidance, encouragement, and support throughout my MSc journey in Digital Innovation and Transformation. His constructive feedback, insightful ideas, and academic mentorship have been instrumental in shaping this thesis. I am especially grateful for the opportunity he provided to access institutional data and for fostering an environment that allowed me to freely explore and develop the necessary skills to complete this work.

My sincere thanks also go to the University of Ottawa DataHub and Brightspace teams—Mr. Todd River, Sébastien Leduc, Sadiki Latty and especially Mladen Dekic and Bensun Fong— for their technical support and collaboration. Their assistance in building a robust data infrastructure, preparing datasets, and facilitating access to institutional platforms was crucial to the success of this project.

I would like to thank my family for their unwavering love and support—especially my parents and sister. Most importantly, I am deeply grateful to my husband, Tin Pham, whose constant encouragement, patience, and belief in me kept me going through the most challenging moments of this journey.

The authors would like to acknowledge the funding for this project provided by the Teaching and Learning Support Service (TLSS) - University Of Ottawa.

All data used in this study were non-identifiable and approved by the University of Ottawa's Research Ethics Board under ethics file number H-04-24-10374.

Dedication

Dedicated to my beloved family

Table of Contents

List of Tables

List of Figures

Abbreviations

ANN Artificial neural network

AUC Area under the receiver-operating characteristic curve

CART Classification and Regression Trees

CNN Convolutional neural network

COVID-19 Coronavirus Disease of 2019

CRISP-DM Cross-Industry Standard Process for Data Mining

DM Data Mining

DSRM Design Science Research Methodology

DT Decision Trees

DW Data Warehousing

EWS Early Warning System

G-Mean Geometric Mean

GaussianNB Gaussian Naive Bayes

GBM Gradient Boosting Machine

GLMNET generalized linear model with elastic net

GNB Gaussian Naive Bayes

GPA Grade Point Average

HEI Higher Education Institutions

KDD Knowledge Discovery in Databases

KNN K-Nearest Neighbors

LA Learning Analytics

LG Logistic Regression

LMS Learning Management System

LR Linear Regression

MCC Matthews Correlation Coefficient

MIL Multiple Instance Learning

ML Machine Learning

MVGP Multi-view Genetic Programming

MVMI Multi-view Ensemble Algorithm

RF Random Forest

ROC Receiver-operating characteristic curve

S3 Student Success System

SIS Student Information System

SMOTE Synthetic Minority Oversampling Technique

SVM Support Vector Machines

SVR Support Vector Regression

U-M The University of Michigan, United States

UBC The University of British Columbia

UoT The University of Toronto

uOttawa The University of Ottwawa

USask The University of Saskatchewan

XGB Extreme Gradient Boosting

Chapter 1

Introduction

1.1 Motivation and Challenges

The transition to remote and hybrid learning—accelerated by the COVID-19 pandemic-has profoundly transformed how students engage with academic institutions. However, this decline in traditional in-person communication is primarily attributable to the shift towards remote learning modalities in response to COVID-19, which has been proven to be accompanied by a significant issue of high dropout rates compared to traditional teaching [1]. On the other hand, the attrition of students from degree programs is a substantial concern for academic institutions due to its adverse effects on student well-being and the broader community. Additionally, it entails financial costs for educational institutions, such as the loss of cash inflows and significant societal costs [2].

Although many higher education institutions have since resumed predominantly in-person instruction, learning activities seem to continue to be extensively mediated through digital platforms. In their analysis of seventeen diverse U.S. institutions using impact-analysis software, which performs virtual experiments on retrospective data, Carmean et al. [3] found that approximately half of the student success initiatives examined, drawing on institutional demographic, academic, enrollment, and engagement data, were already in operation prior to the pandemic and were adapted rather than abandoned during campus closures. This finding indicates that students' digital footprints are not a temporary artifact of COVID-19 but a persistent feature of contemporary higher education operations. Consistent with these findings, Irhouma and Johnson [4] reported that 70% of respondents across Canadian institutions expected an increase in hybrid course offerings, while 59% anticipated growth in fully online offerings. Explaining this shift, Subiyantoro [5]

characterized the pandemic as a catalyst for educational transformation, accelerating the adoption of online learning technologies and pedagogies at scale and concluding that online learning is likely to remain integral to higher education beyond the pandemic period.

Learning Analytics (LA) has therefore emerged as a powerful approach for improving educational outcomes by enabling the systematic collection, analysis, and application of student data to support evidence-based decision-making. During the COVID-19 pandemic, many institutions were not fully prepared for rapid digital transformation and large-scale online teaching, which led to instructors struggling with pedagogical approaches for online delivery and learners requiring timely support and meaningful interaction. The 2022 Canadian Student Wellbeing Study survey [6] revealed that 72% of higher education students experienced a substantial decline in face-to-face interactions with their instructors. Moreover, 40% of students expressed an intention to withdraw from their academic institutions, and 32% reported feeling unsupported by their respective institutions. In this context, Celik et al. [7] suggested that understanding and leveraging the potential of LA tools can provide valuable insights for addressing these challenges and informing the future of higher education policies. Similarly, as evidenced by Carmean et al. [3], their multi-institutional meta-analysis demonstrated that academic programs leveraging data analytics and prediction-based student success scores exhibited clear evidence of intervention efficacy, particularly when fostering collaboration and connection among students, peers, mentors, and academic advisor. Their findings further highlight that the use of analytics to translate institutional data into actionable insights plays a critical role in supporting student success initiatives and guiding institutional decision-making. From higher education policy and management perspectives, Nguyen et al. [8] highlighted that the use of educational technologies can reveal important learning and engagement metrics that were previously overlooked by institutional stakeholders and policymakers. Similarly, after examining the policies implemented during the pandemic period, Mula-Falcón et al. [9] argued that university systems should begin laying the groundwork for sustained educational innovation and invest in training educators to adapt to online and hybrid learning modalities in preparation for future crises.

While LA is gaining significant momentum in higher education institutions, most implementations continue to focus primarily on identifying students at risk of failing [10, p. 13], which could be considered the most accessible and high-impact starting point. At uOttawa, however, individualized interventions based on LA remain largely absent. The university has yet to establish a centralized data infrastructure or implement a formal LA strategy. This lack of an integrated data hub limits the institution's ability to monitor student progress, detect risk factors, and deliver timely academic support, positioning uOttawa as an outlier in the national LA landscape (see section 3.3).

To initiate an institutional LA project, this thesis focuses on applying LA to identify students who experience adverse academic outcomes in a course (A detailed delineation of the criteria used to identify and distinguish at-risk students is provided in the section 6.3.1.1). **Machine Learning (ML)** techniques, which lie at the predictive stage of many LA workflows, offer a promising pathway toward this goal. However, existing literature often emphasizes model performance while neglecting essential stages such as data preparation, feature engineering, and thorough model evaluation. Moreover, few studies validate their approaches within real-world academic settings. Given the variability in institutional data structures and student populations, effective LA implementation demands tailored data engineering and modeling strategies that reflect the specific context. At uOttawa, data from the SIS and the Brightspace LMS remain siloed, making integrated and longitudinal analysis difficult. Additionally, variability in course design, grading policies, and evolving patterns of student engagement further highlight the need for flexible, context-aware, and temporally sensitive predictive models.

This thesis has identified **four key challenges** that must be addressed:

1. **Lack of Comprehensive Research on the Initial Stages of LA Implementation:** There is a noticeable gap in scholarly literature addressing the early phases of LA implementation within HEI in Canada. While many institutions publicly showcase their LA initiatives on official websites—often highlighting current system capabilities or dashboards—these descriptions are typically developed by IT or analytics teams and lack academic depth. They seldom provide detailed insights into the foundational processes, methodology involved in initiating large-scale LA projects. As a result, there is a pressing need for rigorous academic research that systematically explores and documents the initial stages of LA implementation.

2. **Lack of Integrated LA Infrastructure at uOttawa**: The university currently lacks a unified data platform to support LA efforts. Establishing a centralized datahub is a critical first step toward enabling scalable, data-driven educational strategies.

3. **Need for a Tailored ML Approach:** Most existing ML research focuses on improving model accuracy using clean, preprocessed, and ideal datasets. However, applying ML in the context of uOttawa requires working with real institutional data, which can be messy, incomplete, and highly specific to the university's academic environment. Therefore, a customized approach is needed—one that includes appropriate data cleaning, feature selection, and evaluation methods tailored to the structure of courses, programs, and student records at uOttawa. This ensures that

the predictive models are not only accurate, but also practical and relevant to the university's needs.

4. **Overly Narrow Definitions of At-Risk Students and Missed Opportunities for Timely Intervention**: Many LA ML-based models identify at-risk students based solely on final grades, overlooking other important risk signals such as early disengagement or course withdrawal. There is a pressing need for predictive models that incorporate temporal dynamics to enable early and actionable interventions.

1.2 Research Questions and Objectives

This section outlines the research questions that were formulated to address the identified challenges in the previous section 1.1 directly. Table 1.1 frames the research questions and their corresponding objectives.

Table 1.1: Research Questions and Corresponding Objectives

Research Questions	Research Objectives
RQ1: How is LA implemented across institutions, and what are the common practices and benefits, challenges observed?	• Explore LA implementation processes and the current state of LA adoption and their challenges in Canadian HEI. • Summarize examples of effective LA applications to guide this research.
RQ2: Which type of data infrastructure is necessary to support scalable LA implementation at uOttawa?	• Explore uOttawa's current data platform, existing LA system framework and data challenges when implementing LA • Identify and develop a system architecture that is suitable for ML-based LA projects in uOttawa.
RQ3: What are the most effective ML approaches for the early detection of at-risk students using empirical datasets?	• Train and evaluate multiple ML models to compare their performance at predicting at-risk students at different time points. • Identify the best-performing models and techniques that are most suitable for predicting at-risk students at uOttawa.

RQ4: What are the key factors influencing student academic performance?	• Use the best-performing ML model to identify and interpret the most important features that influence student academic risk and performance.

The first research question (RQ1) and its associated objectives directly target challenge 1 in section 1.1 by examining how LA is currently implemented across institutions, particularly in Canada. By systematically identifying common practices, benefits, and challenges, this study brings academic rigor to what is often only described informally on institutional websites or IT documents. Furthermore, by summarizing examples of effective implementations, the research provides a foundation for other institutions—especially those in early LA stages—to build on. This approach ensures that the study does not merely report on existing systems but contributes scholarly insight into the processes and decision-making that underpin successful LA adoption, which is currently underrepresented in the literature.

The second research question (RQ2) is directly aligned with addressing the absence of a unified datahub at uOttawa (Challenge 2- Section 1.1). The corresponding objectives include designing a scalable system architecture tailored for machine learning-based LA efforts. This infrastructure is critical not only for enabling cross-departmental data consolidation but also for supporting long-term, scalable analytics across various educational units. The proposed framework will incorporate multiple data sources—such as SIS, LMS, providing the necessary technical foundation to support advanced analytics and ensure sustainable implementation of LA initiatives at uOttawa.

The third research question (RQ3) and its objectives address the challenge of needing a context-sensitive ML solution (Challenge 3- Section 1.1). Rather than relying on ideal or pre-cleaned datasets, the study explicitly involves training and evaluating multiple ML models on real institutional data. This includes applying feature selection and data preprocessing techniques suited to uOttawa's unique course structures, grading systems, and student demographics. The goal is to identify models that are not only accurate but also robust, interpretable, and practical for the university's actual data environment. This ensures that the predictive solutions developed will be deployable and useful in real academic decision-making, rather than remaining academic exercises with limited applicability.

Finally, the fourth research question (RQ4) focuses on discovering the key factors that influence student academic performance, which enables a more nuanced understanding of academic risk (Challenge 4- Section 1.1). Instead of defining "at-risk" solely by final grades, this research integrates temporal features and behavioral indicators (e.g., early

disengagement, course load, performance trends) to enable early and actionable predictions. By doing so, it moves beyond narrow definitions and allows for timely interventions, such as academic advising or targeted support, that can occur before a student fails a course or drops out. This approach opens the door for more proactive and preventive strategies, which is a key goal of effective LA systems.

1.3 Stakeholders and Objectives

Table 1.2 summarizes all possible stakeholders and their respective objectives within the proposed Learning Analytics framework. Students form the focal point of the system, benefiting directly from early identification of academic risk and personalized feedback. Academic advisors and faculty act on the predictive outputs to deliver timely, targeted interventions that promote student success. Researchers and data analysts are responsible for ensuring the technical robustness and ethical validity of the predictive models. Institutional administrators utilize aggregated analytics to guide policy decisions, resource allocation, and retention strategies. Together, these stakeholders contribute to a cohesive ecosystem that integrates data-driven insights into continuous educational improvement.

Table 1.2: Stakeholders and Their Objectives in this thesis.

Stakeholders	**Objectives and Roles**
Students	• Self-regulate their learning behavior based on early feedback from predictive models. • Seek timely support from academic advisors when identified as at risk.
Academic Advisors / Faculty	• Use model predictions to identify and monitor at-risk students. • Provide personalized interventions, mentoring, or academic adjustments to improve student outcomes.
Researchers / Data Analysts	• Design, validate, and refine Learning Analytics models for accuracy, fairness, and interpretability. • Ensure ethical use and continuous improvement of predictive algorithms.
Institutional Administrators	• Utilize analytical insights to inform policies and optimize resource allocation. • Enhance student retention, reduce attrition costs, and strengthen institutional performance.

1.4 Thesis Contribution

The main contributions of this thesis are both theoretical and practical.

On the theoretical side, the research introduces a conceptual LA infrastructure that integrates data collection, storage, and analysis through a layered data warehouse framework and an adaptation of the CRISP-DM methodology to educational settings. It also defines new constructs, including a refined at-risk student definition and a phase-to-phase predictive model approach.

On the practical side, the study develops and validates an operational data warehouse and data pipeline for ML-based LA, generates new time-based learning features, and implements Python workflows for model training and evaluation. Collectively, these contributions establish a unified, data-driven framework that enhances early risk detection and supports evidence-based decision-making in higher education.

1.5 Thesis Outline

This thesis is organized as follows:

Chapter 2 presents the research design and methodology adopted in this study. It describes the application of the DSRM frameworks, outlining their respective phases and operational processes in chronological order. The research scope, assumptions, and delimitations are also clarified in this chapter.

Chapter 3 addresses Research Question 1 (RQ1) by examining common LA practices adopted by researchers and higher education institutions. This chapter reviews the evolution of LA stages and highlights their institutional benefits through representative applications worldwide. In addition, it analyzes the current Canadian LA landscape, emphasizing adoption trends and institutional maturity levels.

Chapter 4 addresses Research Question 2 (RQ2) by investigating existing LA system architectures and institutional implementation practices. A review of relevant literature is complemented by an audit of data sources at the University of Ottawa, including the SIS, the Brightspace LMS, and the existing Microsoft Azure–based data platform. The analysis identifies limitations in current LA capabilities, particularly within Brightspace's Student Success System (S3), which motivate the design of a more robust and scalable LA architecture. Based on these findings, a context-specific system architecture is proposed to support integrated data management and future analytics for at-risk student prediction.

Chapter 5 presents a systematic literature review to support the development phases required to address Research Questions 3 and 4 (RQ3 and RQ4). The review focuses on the evaluation of machine learning–based prediction models using LA data in higher education institutions, with particular emphasis on empirical studies leveraging LMS and SIS data, while excluding self-reported datasets. The findings inform a gap analysis and provide methodological guidance for the machine learning–based LA implementation described in the subsequent chapter.

Chapter 6 addresses Research Questions 3 and 4 through an empirical investigation of predictive modeling and feature interpretation for the early identification of at-risk students within a Learning Analytics context.

Chapter 7 concludes the thesis by summarizing the key findings and reflections when answering each research question, discussing theoretical and practical contributions, and outlining the study's limitations and directions for future research.

Chapter 2

Research Design and Methodology

This chapter presents the research design and methodology employed in this thesis. It outlines the application of the Design Science Research Methodology (DSRM) and the CRISP-DM framework, detailing their phases and operational processes as applied in this study. The chapter also defines the research scope, assumptions, and delimitations, establishing a structured foundation for the subsequent analytical and empirical investigations.

2.1 Research Design

Research design provides the high-level overall blueprint for how the study addresses its research problems and questions, defining the structure, scope, and approach of the investigation. This study adopts an applied, mixed-methods research design to explore and implement effective ML-based LA solutions for the early identification of at-risk students at the University of Ottawa. The design is structured to systematically answer the research questions presented in Section 1.2, through a combination of exploratory analysis, architectural design, and empirical evaluation.

2.1.0.1 Addressing RQ1: Exploring LA in HEI

To address RQ1, an exploratory research approach is undertaken. This involves an extensive review of existing literature, institutional reports, and public documentation from higher education institutions (HEIs), particularly those that have implemented LA initiatives. The objective is to gain a deeper understanding of the foundational stages of LA

implementation, the associated data challenges, and the demonstrated benefits of LA in practice. Furthermore, a survey of Canadian HEIs is conducted to assess the national landscape of LA adoption and to benchmark institutional efforts.

2.1.0.2 Addressing RQ2: Exploring Existing LA System Architectures and Implementation Best Practices

To address RQ2, this study first investigates existing Learning Analytics (LA) system architectures and common implementation challenges through a review of relevant literature. This is followed by a comprehensive audit of institutional data sources at the University of Ottawa, including both the SIS and Brightspace LMS, as well as an examination of the current Microsoft Azure–based data platform. These activities were supported by consultations with the university's Data Hub team. Furthermore, special attention is given to the current LA capabilities of Brightspace—uOttawa's LMS. Through critical analysis of its features and limitations, particularly the S3, we identify gaps that motivate the development of a more robust prediction model for identifying at-risk students.

The findings from the literature and institutional data review inform the design of a scalable, context-specific LA system architecture for uOttawa. The proposed architecture aims not only to integrate disparate data sources into a centralized data hub but also to support efficient analytics queries and ensure future extensibility.

2.1.0.3 Addressing RQ3 and RQ4: Predictive Modeling and Feature Interpretation

Research questions RQ3 and RQ4 are addressed through an empirical research process. The investigation begins with a review of relevant literature to examine machine learning (ML) methodologies, including best practices for data preparation, model training, evaluation, and interpretability. This foundational review informs the development of a tailored ML prediction model suited to the University of Ottawa's data landscape.

The Institutional data are collected from multiple sources, including the SIS and LMS. These datasets are cleaned, preprocessed, and integrated to support model training. Several classification algorithms are then developed and evaluated using established performance metrics, including accuracy, recall, and the Kappa statistic. This comparative analysis enables the identification of the most effective models for predicting student risk levels at various time points during the academic term.

Beyond model performance, the study places strong emphasis on interpretability—an essential consideration for educational stakeholders. Feature importance analysis is conducted to identify the most influential predictors of student performance, thereby providing meaningful insights into the factors contributing to academic risk. These insights are intended to support timely and actionable interventions for at-risk students.

2.2 Methodology

Guided by the Problem-Centered approach of DSRM, which originally formalized by Peffers et al. [11], the research follows a structured process to develop, demonstrate, and evaluate an innovative artifact for predicting at-risk students. DSRM consists of six iterative phases as illustrated in Figure 2.1: (1) Problem Identification and Motivation, which defines the research problem and establishes its relevance; (2) Definition of Objectives for a Solution, which translates the problem into measurable and achievable goals; (3) Design and Development, where research artifacts are created, including models, methods, data pipelines, or software prototypes; (4) Demonstration, which shows how the artifact addresses the problem in a real or simulated context; (5) Evaluation, which assesses the artifact's effectiveness, quality, and utility using appropriate metrics and validation techniques; and (6) Communication, which disseminates the research outcomes to both academic and practitioner audiences.

Within DSRM, artifacts represent the core research outputs and may include conceptual frameworks, analytical models, algorithms, data schemas, or implemented systems. In the context of LA, artifacts commonly encompass data integration pipelines, feature engineering strategies, predictive models, and evaluation protocols. By emphasizing both rigor and relevance, DSRM ensures that the proposed artifacts are theoretically grounded, empirically validated, and practically applicable.

The following subsections outline how each activity has been applied in this research.

2.2.1 Phase 1: Identify Problem and Motivate

The motivation for this study stems from the institutional need to implement a MLbased-LA system at uOttawa to support the early identification of at-risk students. To frame this research, four key challenges have been identified and formulated as research problems, as detailed in Section 1.1. These problems serve as the foundational drivers for the design and development of the proposed artifacts.

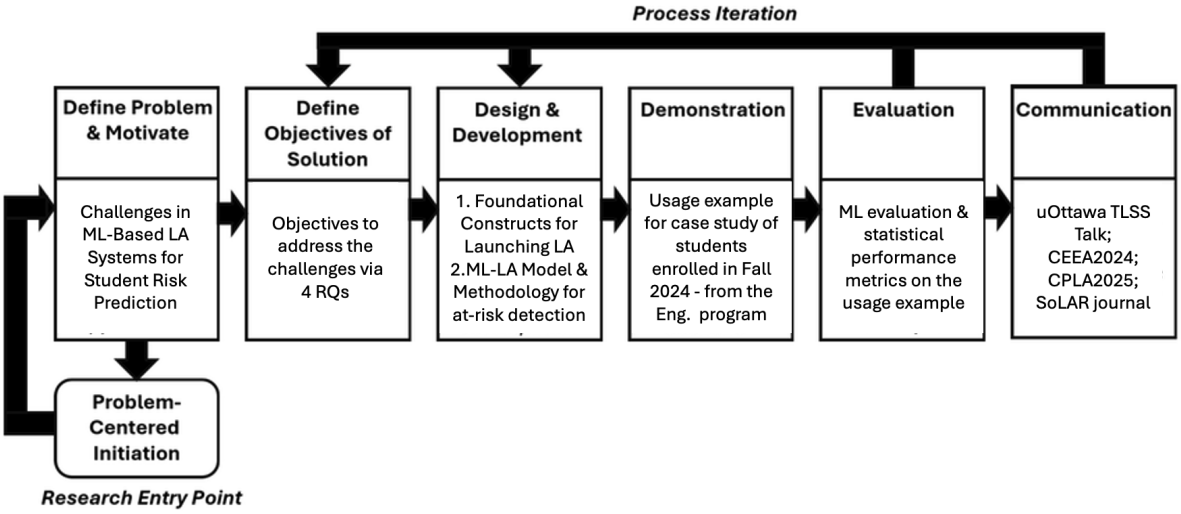


Figure 2.1: DSRM process with Problem-Centered approach in this study. Adapted from the original DSRM in [11]

2.2.2 Phase 2: Define objectives of a solution

Based on the identified research problems, a structured set of research questions and corresponding objectives has been developed, as presented in Table 1.1. This step clarifies the intended outcomes of the research and ensures that the proposed artifacts are designed to directly address the institutional and academic challenges previously identified. A detailed explanation of how each objective aligns with the research problems is provided in Section 1.2.

2.2.3 Phase 3: Design and development

In this phase, a set of artifacts is designed and developed to fulfill the research objectives. These artifacts include constructs, models and methods tailored to address the research questions and solve the stated problems. A summary of the designed artifacts is outlined in Table 2.1.

Table 2.1: Summary of Research Contributions as DSRM Artifacts and Their Corresponding Research Questions

Artifact Type	Artifact Description	RQ
Construct	A critical review of LA adoption strategies and successful applications in global HEI, with a specific focus on Brightspace, the current LMS used at uOttawa. This construct identifies stages of LA maturity, explores existing DW frameworks supporting LA, and highlights institutional challenges in early-stage implementation. It also analyzes the current state of LA adoption and its limitations, providing theoretical grounding to align LA practices with our strategic goals for identifying and predicting at-risk students at uOttawa.	RQ1, RQ2
Model	A scalable institutional LA system architecture tailored for uOttawa, which integrates data from the SIS and LMS into a unified pipeline that supports machine learning–driven predictive tasks. The architecture also embeds a ML model for at-risk student prediction and defines a comprehensive data flow encompassing data extraction, transformation, prediction model training, and evaluation.	RQ2, RQ3
Method	A methodological framework for data preparation, model validation, evaluation, and comparing machine learning algorithms used to predict at-risk students at four academic checkpoints (start, 30%, 50%, and 75% of the course timeline). The method emphasizes phase-specific model performance, with evaluation metrics including recall, accuracy, and Cohen's Kappa score. It also incorporates temporal analysis of feature importance to capture how predictive factors evolve throughout the term.	RQ3, RQ4

Regarding the artifact development, except for the designed Construct artifact which required an exploration research as detailed in section 2.1 (Addressing RQ1&RQ2), the rest artifacts (Model and Method) required a Data Mining (DM) process which are guided by the CRISP-DM- a framework to ensure a structured and transparent research process with extensibility in mind to support future enhancements. CRISP-DM builds upon and elaborates the original Knowledge Discovery in Databases (KDD) framework, which Fayyad et al. [12] defined as the overall process of transforming large volumes of raw data into valid, novel, useful, and understandable knowledge. While data mining plays a central role, KDD also emphasizes important steps such as data preparation, selection, cleaning,

incorporation of prior knowledge, and proper interpretation of results to derive meaningful insights. CRISP-DM further extends this approach by organizing the process into six interconnected phases: Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment [13]. During the past two decades, this methodology has become widely adopted and is often regarded as a "de facto standard for data mining projects," particularly in the health and education sectors [14].

The following sections will detail each phase of the CRISP-DM methodology, with the exception of the Deployment phase, which is beyond the scope of this thesis and will be addressed in future work. Figure 6.1 depicts the six steps of the CRISP-DM framework and illustrates their adaptation to the context of early prediction of at-risk students in higher education. The phases in CRISP-DM are not strictly linear; moving back and forth between them is often necessary, depending on the outcomes of each step [15]. The coloured arrows in Figure 6.1 represent common dependencies, while the grey arrow and circle reflect the cyclical nature of data mining—where lessons learned from one project inform and improve future iterations.

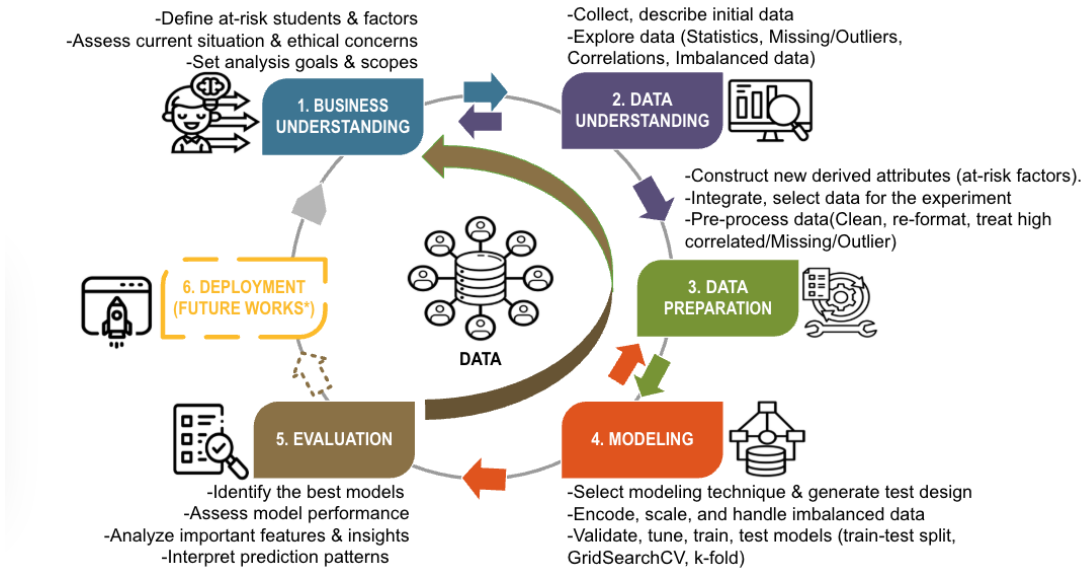


Figure 2.2: Adapted CRISP-DM Framework For At-risk Student Prediction Project (Deployment phase excluded from current study and will be addressed in future work). Adapted from CRISP-DM guidelines [15]

2.2.3.1 Business Understanding

This initial phase focuses on identifying the project's goals and requirements from an institutional perspective and translating them into a clearly defined data mining problem [15]. In this study, the Business Understanding phase was informed primarily by the author's analysis of University of Ottawa academic regulations, including grading scales, definitions of course failure, and withdrawal policies, which were used to operationalize the concept of at-risk students. In parallel, this phase involved collaboration with the institutional data hub team to define the scope, granularity, and availability of relevant student data across existing information systems. These discussions informed both the feasibility of the proposed LA approach and the identification of candidate data features expected to support subsequent predictive modeling. The resulting problem formulation and requirements are detailed in Section 6.3.1. In addition, one of the guiding requirements of this study is to ensure that the proposed methodology remains reusable and extensible for future research, as supported by the LA architecture introduced in our first paper.

2.2.3.2 Data Understanding and Data Preparation

The Data Understanding and Data Preparation phases involve exploring, cleaning, transforming, and organizing raw data into a structured format suitable for ML modeling [15]. Unlike many ML studies that focus primarily on the modeling phase, this research emphasizes the foundational role of data preparation. These two phases are interwoven and iterative, supporting two key objectives stated in the previous business understanding step: (1) designing and developing a data pipeline aligned with the proposed LA architecture, and (2) preparing high-quality data for ML modeling in the subsequent phase. Table 2.2 outlines the specific operational steps, techniques, and tools employed at each stage of the workflow.

Table 2.2: CRISP-DM Data Understanding & Preparation: Chronological Operational Processes and Techniques

Operational Process / Objectives	**Methods / Techniques**	**CRISP-DM Phase**
Step 1: Initial Data Overview Understand and document the structure, content, and quality of raw institutional data from multiple sources (e.g., SIS and LMS).	Document table structures, data types, and relationships using Visio. Create a high-level data dictionary for reference.	Data Understanding
Step 2: Design the Data Warehouse Model Reorganize raw transactional data into an analytics-friendly dimensional structure for downstream use.	Design a star-schema model based on data warehousing best practices in [16], consisting of fact tables (e.g., student outcomes) and dimension tables (e.g., time, course, student).	Data Understanding
Step 3: Develop ETL Pipelines Extract, clean, and consolidate data from SIS and LMS systems into a unified data mart within the data warehouse.	Implement ETL processes using SQL scripts to execute data extraction, transformation (e.g., table joins, at-risk label mapping), and loading into the DW layer.	Data Preparation
Step 4: Feature Engineering and Dataset Filtering Generate derived features to capture student behavior and engagement over time, and prepare the dataset for modeling.	Create temporal derived features (see Table 6.3). Filter out irrelevant or incomplete records to ensure quality inputs for model training.	Data Preparation
Step 5: Descriptive Data Analysis Summarize and visualize distributions, frequencies, and central tendencies of each variable to detect irregularities.	Calculate count, mean, median, min, max, standard deviation, and quartiles using pandas (Python). Visualize with histograms, boxplots, and bar charts.	Data Understanding
Step 6: Correlation Analysis Examine relationships between numerical and categorical variables to detect multicollinearity and dependencies.	Use Pearson's correlation coefficient for continuous variables [17], and Cramér's V for categorical associations [18]. Visualize with heatmaps and correlation matrices.	Data Understanding

Operational Process / Objectives	Methods / Techniques	CRISP-DM Phase
Step 7: Outlier and Rare Category Detection Identify statistical outliers and infrequent values that may affect model performance.	Apply the Interquartile Range (IQR) method [19] to detect numeric outliers. Use frequency counts to flag categorical values with extremely low occurrence.	Data Understanding
Step 8: Address Highly Correlated Features Reduce redundancy and potential overfitting by removing correlated features.	Drop one feature from each pair with correlation above a defined threshold (e.g., $\|r\| > 0.8$) using a heatmap and pairwise correlation matrix.	Data Preparation
Step 9: Handle Missing Values Impute or clean missing data to maintain model integrity and prevent bias.	Use mean imputation for numerical features. Apply placeholder categories like "Unknown" for missing categorical values. Implement using pandas or `SimpleImputer` in scikit-learn.	Data Preparation
Step 10: Treat Outliers Cap or replace outliers to reduce noise and improve model robustness.	Clip numerical values at IQR bounds (Q1 - 1.5×IQR, Q3 + 1.5×IQR). Reclassify rare categorical values into "Other" group.	Data Preparation

2.2.3.3 Modeling & Evaluation

This phase initiates the development of predictive models for identifying at-risk students. Multiple ML methodologies and techniques, compatible with the Python environment used in this thesis, are applied and described in detail. Table 2.3 outlines each modelling and evaluation step, including the corresponding methods and techniques employed.

Table 2.3: CRISP-DM Modeling & Evaluation: Chronological Operational Processes and Techniques

Operational Process / Objectives	Methods / Techniques	CRISP-DM Phase
Step 11: Identify potential prediction algorithms Select suitable machine learning algorithms to be used in the experimental phase within the Python environment.	Six classifiers are selected: RF, LG, SVM, and GaussianNB from the Scikit-learn library, and XGB from the XGBoost library. The rationale for selecting these algorithms is discussed in Section 6.3.3.1	Modeling
Step 12: Encode Categorical Features Convert categorical variables into a machine-readable format compatible with most classification models.	Apply One-Hot Encoding using `OneHotEncoder` from `scikit-learn`, which transforms each categorical feature into a binary column, preserving non-ordinal relationships without imposing artificial ordering.	Modeling
Step 13: Split Data for Model Validation Partition the dataset into training and testing subsets to enable unbiased evaluation of model generalization.	Use an 80:20 split strategy for train:test datasets, implemented via `train_test_split()` from `scikit-learn`.	Modeling
Step 14: Normalize Features Standardize numerical features to ensure consistent scaling across predictors, which is crucial for distance-based and gradient-sensitive models.	Use `StandardScaler` from `scikit-learn` to scale features by removing the mean and scaling to unit variance. This step is applied only to non-tree-based models like LG, GaussianNB and SVM.	Modeling
Step 15: Hyperparameter Optimization Fine-tune model performance by searching over predefined combinations of hyperparameters.	Apply hyperparameter- tuning [20] using `GridSearchCV` [21] with 5-fold stratified sampling to explore optimal settings for model parameters.	Modeling
Step 16: Handle Class Imbalance Mitigate bias introduced by imbalanced class distributions, which is identified at step 5.	Use built-in imbalance handling parameters in tree-based classifiers, and apply SMOTE from the `imblearn` package for models that lack native support for imbalance handling.	Modeling

Operational Process / Objectives	Methods / Techniques	CRISP-DM Phase
Step 17: Model Training Train and compare multiple supervised ML algorithms on the prepared training set.	Train selected models using `scikit-learn` and `xgboost` libraries. Validate and evaluate during training using cross-validation in GridSearchCV [21] and store the trained champion model for later deployment.	Modeling
Step 18: Model Evaluation Assess the performance and reliability of trained models using classification and statistical metrics.	Evaluate predictive accuracy, recall. Use Cohen's Kappa to assess inter-rater agreement between predicted and actual classes.	Evaluation

2.2.4 Phase 4: Demonstration

To demonstrate the applicability and effectiveness of the developed artifacts, a case study is conducted using institutional data from the Bachelor of Applied Science in Engineering program at the uOttawa. The focus is on students registered in the Fall 2024 term. This demonstration showcases how the integrated LA system and predictive models function in a real educational context.

2.2.5 Phase 5: Evaluation

The evaluation process assesses the utility, accuracy, and interpretability of the developed artifacts. The institutional data hub feeds into the ML prediction model, highlighting the importance of reliable data integration for model effectiveness. The evaluation includes both standard ML performance metrics (e.g., accuracy, recall, F1 score) and statistical validation using Cohen's Kappa to assess prediction reliability. Further details on the evaluation criteria, datasets, and comparative methods are provided in the section "Research Process and Methods."

2.2.6 Phase 6: Communication

In accordance with DSRM, the Communication phase focuses on disseminating the research problem, designed artifacts, and evaluation results to relevant academic and practitioner audiences. This includes presentations at academic venues and targeted journal submissions aimed at advancing Learning Analytics research and practice. This work has been presented at the Teaching

and Learning Support Service (TLSS) at the University of Ottawa and accepted at the Canadian Engineering Education Association Conference (CEEA-ACÉG 2024) and the Cyber-Physical Learning Alliance Summit (CPLA 2025). Building on feedback from prior dissemination efforts, the journal manuscript is being refined to strengthen its alignment with Learning Analytics and educational data mining literature prior to submission to an appropriate peer-reviewed journal.

2.3 Research Scope and Delimitation

This study focuses on designing, developing and evaluating a ML-based LA model for predicting at-risk undergraduate students at the University of Ottawa. The model utilizes academic data drawn from the SIS (uOcampus system), covering student records from Fall 2019 to Fall 2024, and from the Brightspace LMS, limited to Fall 2024 courses that that made use of the this online platform, specifically those in which content was delivered through Brightspace before the 80% completion point of the course duration. Courses that do not use the Brightspace LMS, and for which no recorded LMS data are available, are excluded from this study. This includes courses that rely primarily on alternative platforms, such as Microsoft Teams or other external tools. The exclusion is necessary because, at present, there is insufficient infrastructure and institutional support from the Data Hub team to reliably retrieve and integrate data from these decentralized and non-official systems.

The research is guided by the DSRM framework, with the modeling component following the CRISP-DM methodology, specifically within the Design and Development phase of the DSRM process. Six selected machine learning algorithms—including LG, RF, SVM, GaussianNB, Decision Trees (DT), XGB—are implemented and evaluated using Python, based on real institutional data.

This study is limited to solely academic data available at the time of writing, which was provided by the uOttawa Data Hub team. It does not assess the downstream impact of interventions following risk identification, nor does it deploy the system in a live institutional setting. The scope is limited to undergraduate students, excluding graduate-level data. Furthermore, only a targeted subset of ML algorithms is evaluated; advanced techniques such as deep learning models, including artificial neural networks (ANNs), etc., are outside the scope of this work. This research also does not aim to develop a fully integrated LA system; instead, it focuses on the design of a LA dataflow architecture and the ML–based prediction component. The development of a complete system—including automated workflows and a user interface component integrated with existing uOttawa data platforms, requires further investigation and is outside the scope of this thesis.

Chapter 3

Learning Analytics Implementation Across Institutions

This section addresses Research Question 1 (RQ1), which examines common Learning Analytics (LA) practices adopted by researchers and institutions. This exploration contributes to the existing literature by providing insights into the evolution of LA stages and their benefits to institutions through sample applications worldwide. Additionally, we explore the current Canadian LA landscape, highlighting the application trends and current stages in implementing LA across institutions.

3.1 Motivation

In the current dynamic landscape of technology integration in education, the obstacle to accessing electronic information generated for and essential to learning experiences has diminished, if not completely vanished. LA, the measurement, collection, analysis, and reporting of data about learners and their contexts, has emerged as a powerful tool to understand and optimize institutional decision-making and enhance student success. While LA is gaining considerable momentum in higher education institutions, the focus is still predominantly on students at risk of failing [10, p. 13].

The motivation behind this RQ1 stems from the increasing importance of LA in improving educational outcomes and institutional effectiveness. As technology continues to play a crucial role in education, institutions need to harness the power of LA to enhance student success. By understanding the challenges and opportunities associated with LA implementation, institutions can better leverage this technology to meet the diverse needs of their students and faculty.

One of the key challenges in implementing LA is the lack of comprehensive research focusing on the foundational stages of LA, especially in the context of Canadian universities. There is a need for more detailed research on its implementation stages, particularly in the early stages of establishing a large-scale LA project. This section aims to fill this gap by providing a thorough understanding of LA implementation and its benefits at different stages among institutions and an overview of the current Canadian LA landscape at scale.

3.2 Background: LA Implement Stages and Challenges

In recent years, the emerging field of LA has been developed to tackle the issue of improving educational outcomes by offering potential solutions. Francis et al. defined LA as "the collection and analysis of the demographic, behavioral, and digital trace data of students to improve their experiences and outcomes by enabling targeted real-time interventions with particular cohorts and individuals based on their profile derived through machine learning and algorithmic processing" [22] LA is an engine that works in five steps: capture, report, predict, act, and refine [23, p. 16]. This process is demonstrated in Fig 3.1

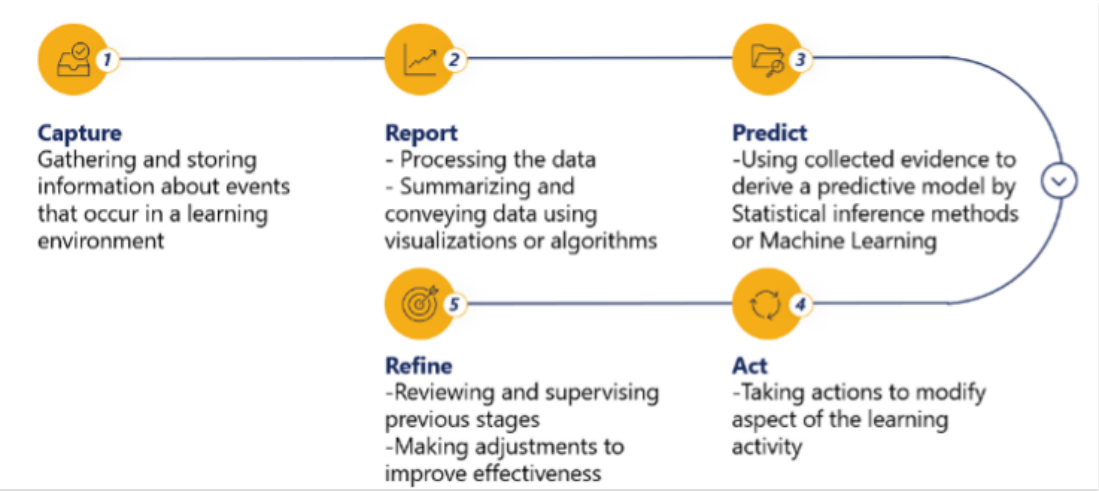


Figure 3.1: Learning Analytics (LA) Stages. Source: authors

LA can support institutional management in several ways, including improving student enrollment, aligning services with institutional goals and outcomes, informing education policies, and providing a more comprehensive view of the institution as a whole [24] Various research studies on LA have been undertaken in HEI globally, covering various topics, including understanding students' behaviors in online learning systems, developing predictive models for student

outcomes, and applying LA methodologies. However, most of the studies analyzed in this context focus on predicting student attrition, which refers to the decline in the number of students attending courses over time, including both dropout and desertion. [10]. Although LA has been recognized as having the potential to bring significant benefits to HEI, its implementation also poses a notable challenge. Research surveys [25] have asserted that data tracking and collection is among the most significant hurdles to overcome when applying LA in practice. The availability of resources is a crucial consideration for educators, as it can affect the feasibility and accuracy of the data collection process. This viewpoint is shared by another survey , [26], which also highlights the issue of data access as a significant barrier that hinders HEI from improving the performance of their LA models. Moreover, the historical nature of the data also creates difficulties for applying LA in HEI. Nurhadi et al. reveal some models are constructed based on static data, and any modifications made to the database can impact the accuracy of the results [26]. This underscores the importance of ensuring that LA models remain up-to-date and that the data used to inform them are continuously refreshed. HEI also face obstacles in collecting and organizing data for the purpose of analytics and visualization. This process can be daunting, given the complexity of the data involved and the need to ensure that it is both relevant and reliable [27].

3.3 LA Implementation Across Institutions

In order to gain insights into emerging trends and innovations in LA from other institutions, this section outlines successful practices from international institutions. Additionally, it examines the current Canadian LA landscape through a survey of 19 Canadian universities.

3.3.1 Survey Overview and Methodology

The field of LA emerged around 2011 [28] and has experienced continuous growth in research interest, particularly since 2017 [29]. Several literature reviews have meticulously examined and summarized various aspects of LA research. These reviews cover topics such as the benefits and challenges of LA ([25], [26]); development and trends across institutions worldwide ([28], [30]); [29]); student data features and source types used in LA ([10]); data mining and machine learning algorithms used in predictive LA [31]; and the types of stories generated by dashboards using predictive analytics [32].

However, many of these surveys focus on small-scale implementations, often at the level of individual courses or limited student cohorts, and therefore provide limited insight into institution-wide adoption and integration of LA practices. In contrast, large-scale LA implementation requires coordinated participation among students, instructors, administrators, and technical teams across an entire institution. To address this gap, the present survey aims to examine the status

of LA implementation at an institutional scale, with a particular focus on Canadian universities. Given that not all institutions publicly disclose detailed information regarding their LA initiatives, this survey adopts a pragmatic and exploratory approach, including universities for which verifiable information on LA projects, usage guidelines, or implementation strategies is publicly accessible. Data are collected primarily from official university websites and supplemented by relevant academic publications. While this approach does not guarantee exhaustive coverage of all Canadian institutions, it supports the construct validity of the survey by grounding the analysis in documented and observable institutional practices, thereby offering a realistic view of the current landscape of large-scale LA implementation.

3.3.2 LA Sample Practices

Among the LA stages (as described in Fig 3.1), the first three stages are the most studied in the LA domain. In contrast, the action stage represents the most significant challenge for implementing and measuring effective techniques. [33, p. 147-166]. In addition, detailed research focusing on the foundational stage of LA is limited. This section highlights international LA sample practices, focusing on the two most common stages: Report and Predict. Researchers have employed various visualization techniques in the second stage of LA to convey different data types. For instance, [34] presents a data dashboard using simple tables to showcase student feedback on course teaching and evaluation. The dashboard features rows for different courses categorized by departments and columns for various types of feedback, such as "The course is well organized," "The workload was reasonable," etc. Users can expand each department to view detailed course feedback. Similar tables are used for learning outcomes and teacher ratings. Additionally, a correlation chart (refer to Fig. 16 in [34]) with a threshold score of 4.5 identifies departments, courses, or teachers that need improvement or redesign to improve student outcomes. On the other hand, to analyze attrition and retention in each department [35], a ribbon chart (Fig 3.2) allows faculty staff to see how groups of students flow between programs throughout their time as students, including students who leave the university without completing their programs. This analysis identifies demographic groups of students in an academic program whose paths differ noticeably from other groups, which can point to specific points in the program that may need more attention.

In [25], the author uses boxplot charts (Fig 3.3) to visualize how students are engaged, measured by the number of views and timing of access to the mandatory lecture notes and study guide. This analysis may be a key resource related to higher quiz scores, and students who accessed this resource in advance had a better quiz score than those who accessed the material closer to the deadline.

In the third stage, the prediction of LA (as described in Fig 3.1), various techniques and methods are available for creating predictive models, and selecting the most suitable one depends on the specific environment and context of the application. [23, p. 29]. Most of the studies

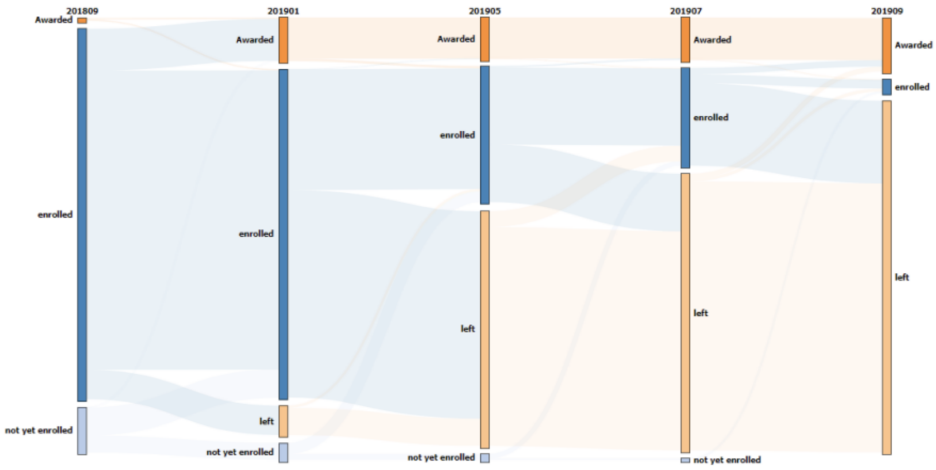


Figure 3.2: Sample of a Ribbon chart: a large portion of students left after the 201901 (the 2nd term from the left)[35]

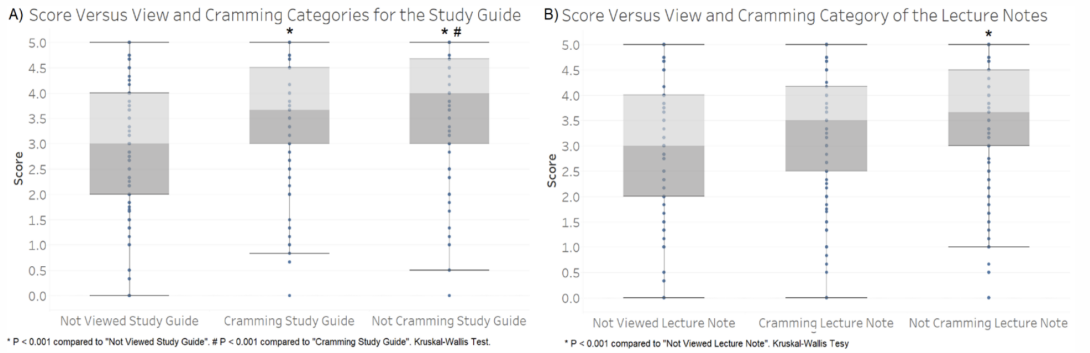


Figure 3.3: Boxlot charts show that students who were Not Cramming the Study Guide had the highest median score, followed by students who were Cramming the Study Guide, and students that were not viewing the material[36]

analyzed concentrate on predicting student attrition, which is the decrease in the number of students attending courses over time, including both dropouts and those who leave abruptly [24, p. 13] . For instance, [37] presents a multiview at-risk student early warning system using comprehensible genetic programming classification rules. This system targets underrepresented and underperforming student populations by integrating multiple student information repositories. It improves prediction accuracy and timing through multiview learning. Through three interfaces, the system offers personalized and aggregated feedback to students, instructors, and staff. Refer

to Fig 3.4 from [37], an example is the student interface dashboard, which allows students to view their performance and compare it with peers anonymously. At week 11 of the course progress,

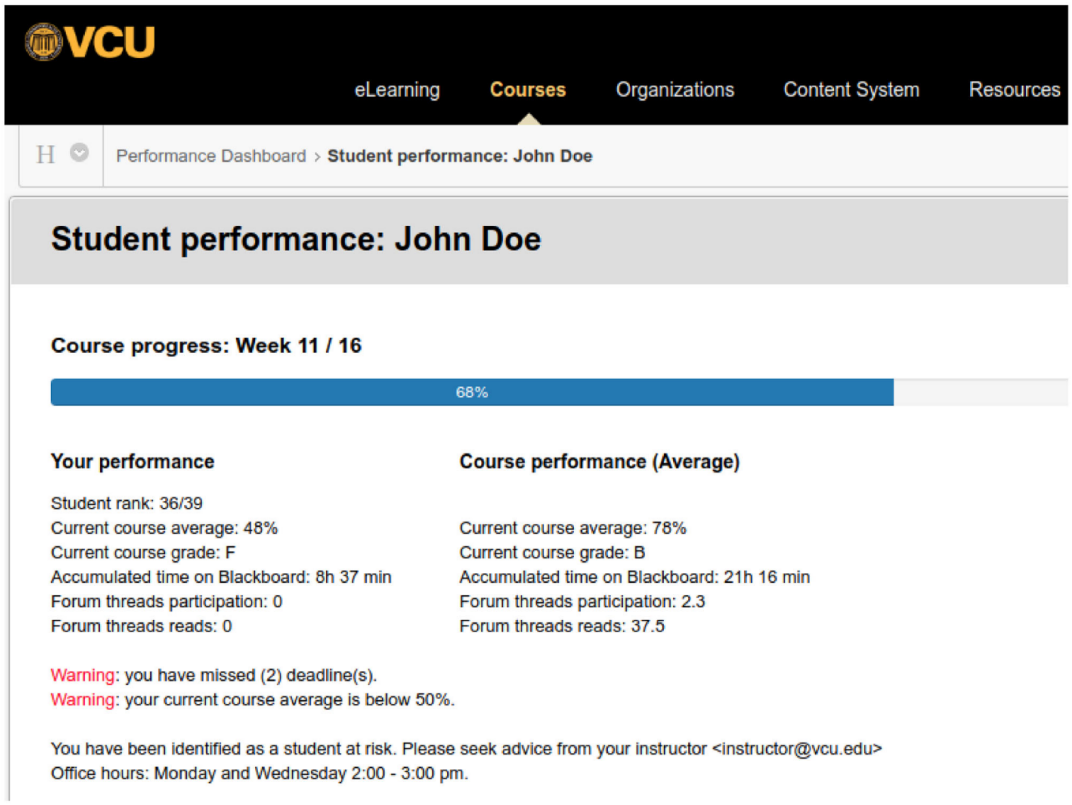


Figure 3.4: Student Dashboard Sample [37]

students receive a ranking based on their current grade to encourage improvement. At-risk students are prompted to contact the instructor for support. Both students and instructors receive email alerts for performance alerts, but the specific rules are not disclosed to prevent cheating.

3.3.3 LA Successful Implementation At Scale

Despite the vast amount of LA research increasing each year since 2018 [38], only a limited number of universities have successfully implemented LA on a large scale.

The Myla project, an open-source software developed at The University of Michigan, United States (U-M) in collaboration with developers at the The University of British Columbia (UBC)[39],

has emerged as a prominent implementation in North America. Myla is a student dashboard that supports adaptive motivation and self-regulated learning by allowing students to compare their performance and behaviors with peers across different grade levels. This comparison helps students decide on a suitable learning path for achieving better learning outcomes. The dashboard features three main views: the course resource access view, which shows how classmates study and what resources they use; the assignment view, which helps students track their scores and plan learning goals based on assignment grades; and the grade distribution view, which includes a histogram of classmates' scores to provide insights into overall performance compared to other students in the class [40]. Since its inception in the fall of 2018, over 7200 students have utilized this tool to enhance their learning experiences. Through pilot testing, Myla received positive feedback from 66% of students, indicating its effectiveness in supporting their learning journeys. Moreover, 72% of students expressed satisfaction with the tool, and an overwhelming 92% expressed a desire for the dashboard to be available in all their courses. [39].

In Canada, UBC and The University of Toronto (UoT) have focused on larger-scale initiatives to benefit students and other stakeholders. UBC launched twelve LA pilots in September 2017 under the LA Project, which aims to empower students, faculty members, and departments with analytics to support teaching, learning, student success, and program planning. At the time of this work, 7 projects were ongoing, 4 were discontinued or on hold, and 1 was completed. UBC is actively seeking contributions to its LA pilot projects, with support provided by the LA project team and available data and tools [41].

UoT, by 2021, had published an LA Strategy Paper outlining a framework for LA capacity development. The strategy emphasized three key opportunities: Evaluate learner engagement/performance in course activities to improve pedagogical design; Provide individualized and tailored feedback to improve student academic success; Enable access to data for academic program planning, advising & coaching [42]. With that in mind, UoT launched five major LA projects in 2022 [43]. On the other hand, the The University of Saskatchewan (USask) has developed seven LA dashboards integrated with its Canvas LMS. These dashboards are specifically designed to support individual instructors and program design at the university [35]

As shown in Figure 3.5, approximately 20% of other Canadian universities surveyed (University of Alberta [44], Simon Fraser University [45], York University [46], Carleton University [47]) have integrated LA tools using existing resources within their LMS to accelerate the adoption of LA practices. Conversely, half of the universities in the survey reported either planned projects only (Western University [48], Queen's University [49], and our University Of Ottawa) or no project information. While these universities have yet to provide any statistical results demonstrating the effectiveness of LA, institutions such as UBC, UoT, and USask have taken proactive steps by setting clear understandings and implementing thorough strategies with action plans to enter the LA race. Despite the limited sample size of universities in the survey, the fact that more than 50% have already implemented or planned to implement LA tools underscores the growing recognition of the need for LA to enhance academic performance and student success within these

Canadian Univesity	LA use	Status as of 2024/02	LMS platform
Carleton University	Yes	Utilized LA features in Brightspace LMS	D2L Brightspace
Queens University	Yes	Planned LA program in 2014, but no published project found on university website	D2L Brightspace
Simon Fraser University	Yes	Utilized LA features in Canvas LMS	Canvas
University of Alberta (UoA)	Yes	Utilized LA features in Moodle LMS	Moodle
University of British Columbia (UBC)	Yes	12 LA projects (since 2017)	Canvas
University of Ottawa	Yes	1 project (2023)	D2L Brightspace
University of Saskatchewan	Yes	7 LA Projects	Canvas
University of Toronto (uOT)	Yes	5 LA projects (since 2022)	Canvas
Western University	Yes	No formal LA currently. Preliminary work has started, but more is needed to build a robust system	D2L Brightspace (2023)
York University	Yes	Utilized LA features in Moodle LMS	Moodle
Concordia University	N/A	N/A	Moodle
Dalhousie University	N/A	N/A	D2L Brightspace
McGill University	N/A	N/A	D2L Brightspace
McMaster University	N/A	N/A	D2L Brightspace
University Of Calgary	N/A	N/A	D2L Brightspace
University of Guelph	N/A	N/A	D2L Brightspace
University of Manitoba	N/A	N/A	D2L Brightspace
University of New Brunswick	N/A	N/A	D2L Brightspace
University of Waterloo	N/A	N/A	D2L Brightspace

Figure 3.5: Survey: Implementation Status of LA in Canadian Universities. Source: authors.

institutions. This trend indicates a strong inclination towards adopting LA practices to improve educational outcomes and institutional effectiveness across the Canadian higher education landscape.

3.4 Summary and Discussion

In the rapidly evolving landscape of technology in education, barriers to accessing electronic learning materials have significantly decreased. Despite the challenges in understanding the implementation and benefits of LA, especially in the early stages of establishing an LA project at a university, this study offers a comprehensive insight into these aspects. A thorough understanding of LA implementation and its benefits at different stages among institutions, along with an overview of the current Canadian landscape of LA, was provided.

Initially, to illuminate the power of LA, we delve into its evolution and showcase real-world benefits through global university applications. The adoption of LA is increasing among insti-

tutions in Canada and worldwide. While some LMS have implemented predictive models and dashboards, limitations like scope remain. Projects like Myla have shown high student satisfaction, highlighting the potential of LA to positively impact the educational landscape in the growing digital learning environment.

From the author's perspective, the limited adoption of institutional LA is not primarily due to a lack of analytical techniques, but rather to organizational readiness and alignment. Universities often approach LA as a technical problem, whereas its success depends on coordinated data governance, clearly defined use cases, and integration with academic support processes. Without a well-structured data infrastructure and agreed-upon intervention pathways, the value of large-scale data analytics remains difficult to demonstrate. Consequently, incremental and context-aware approaches, grounded in existing institutional systems and policies, may represent a more realistic pathway toward sustainable LA adoption in higher education.

Chapter 4

Brightspace's Direction and A Proposed System Design for LA

This chapter addresses Research Question 2 by examining existing LA system architectures and institutional implementation practices. A review of prior literature is complemented by an audit of data sources at the uOttawa, including the SIS, Brightspace LMS, and the current Microsoft Azure–based data platform. The analysis highlights limitations in existing LA capabilities, particularly within Brightspace's Student Success System (S3), which motivate the need for a more robust and scalable LA architecture. Based on these findings, a context-specific architecture is proposed to support integrated data management and future analytics for at-risk student prediction.

4.1 Motivation

Despite the growing body of LA research, many HEI continue to face difficulties in implementing LA at an operational and institutional scale, as evidenced by the survey results presented in Section 3.3. Existing studies predominantly emphasize predictive models and analytical techniques (a summary of previous survey works is provided in Chapter 5, Section 5.2), while offering limited guidance on system architectures capable of addressing fragmented institutional data, evolving academic policies, and privacy constraints. At the University of Ottawa, student-related data are distributed across multiple platforms, including the SIS , the Brightspace LMS, and cloud-based data services, leading to siloed data access and limited analytical interoperability. In addition, LMS-provided analytics tools, such as Brightspace's Student Success System (S3), which will be detailed in 4.3, provide restricted customization and limited support for institution-specific definitions of academic risk. Together, these challenges motivate a systematic examination of

LA system architectures and implementation practices, with the objective of designing a scalable and context-aware infrastructure that can effectively support the early identification of at-risk students.

4.2 Background: Data Warehouse (DW) Infrastructure for LA

To address these LA challenges, the focus should shift towards how to collect and manage the large volume of historical data so that these resources can be re-used and accessed effectively and always up to date. Data warehousing serves as a suitable mechanism to support these endeavors. glsdw is a process of accumulating vast amounts of data from multiple sources to create a huge data repository exposed to ad hoc business queries to produce required information within an acceptable response time [8]. It involves identifying crucial business activities and data sources, designing the process for extracting, transforming, and loading data (ETL), developing a schema, creating data aggregation, designing metadata, determining the indexing strategy, planning the architecture, and choosing tools for accessing the data (Fig 4.1).

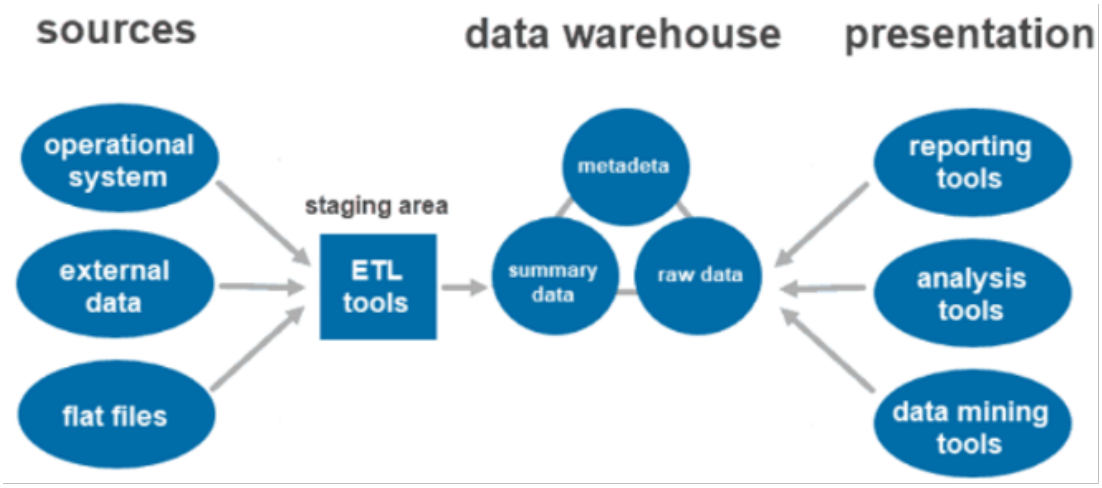


Figure 4.1: A traditional Datawarehouse architecture [50]

In the LA domain, Li et al. [34] have proposed a novel conceptual approach for creating a teaching and learning data warehouse analytics system that utilizes a multi-dimensional analysis approach (as shown in Fig 4.2). This involves storing student data from a LMS into different layers of a data warehouse and categorizing it into different key areas, including applicant profiles, student enrollment, academic performance, graduate employment, student admission,

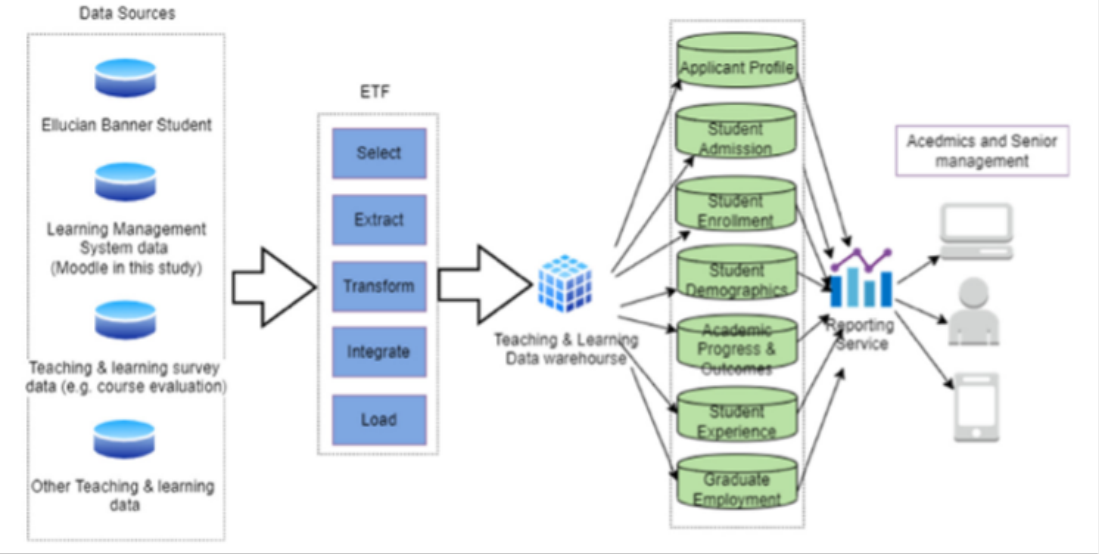


Figure 4.2: A Proposed Data Warehouse for LA in[34]

student demographics, and student experience. Data cubes are used to analyze each area in different dimensions, such as courses, academic year, and student personal information. The research describes a clear data flow through each layer of data, including data sources, extract-transform-load, data warehouse, and analysis. Multiple analytical examples have demonstrated the effectiveness of this approach in [34]. The study efficiently addressed the data-related challenges in the capture and report stages of LA and developed a robust data infrastructure for further research on LA stages, excluding the predict and act stages due to the study's scope.

Recent studies on LA have also employed the data warehouse approach, with some proposing advanced concepts beyond the conventional data warehouse, such as the use of big data [51] and data lake [52] in implementing prediction models with machine learning or advanced AI techniques, including deep learning or neural networks. Yet, due to their high complexity, these studies only suggest data architecture and call for future research to utilize these data without providing proof of practical efficacy.

On the other hand, a study proposes a solution to minimize the complexity and cost of implementing a practical LA project, resulting in an accurate prediction model and effective intervention protocol, with more than 7% of students passing the same course pilot study [53]. Choi et al. opted for a different approach by bypassing the data warehouse and using free cloud services to gather data from Clicker, a device used for collecting student responses, surveys, or performance data. The collected data is then fed into a Google Sheet cloud service where prediction algorithms are implemented to identify at-risk students at different stages of the course. Despite using a small dataset, positive accuracy scores were reported. At-risk students are then

simply provided with a comprehensive intervention protocol, including talking with instructors and receiving reminders via email or warning. Although this study focuses solely on the prediction models and ignores the capture and report stages, the goal of improving student performance is still achieved without the need for a data warehouse. Despite that, this study might not be considered a LA project since no student pattern analysis could conduct to understand the hidden reasons why students struggle.

With regard to technical considerations for configuring a data warehouse, there are two solutions that require careful consideration: on-premises and cloud-based data warehousing. While on-premises data warehousing represents the traditional approach, cloud-based solutions are managed and hosted by third-party cloud service providers. This affords the benefits of inherent flexibility in a cloud environment, coupled with more predictable costs that can be based on usage or a fixed amount. Furthermore, the up-front investment for cloud-based data warehousing is typically much lower, and lead times are shorter compared to on-premises solutions, as the need to purchase hardware is eliminated. In light of these factors, an increasing number of companies are transitioning from traditional data warehousing to cloud-based solutions, taking advantage of the cost savings and scalability afforded by managed services [54]. Nevertheless, there is a limited number of researchers who utilize cloud services for implementing LA projects have found during this thesis study.

4.3 DL2 Brightspace Learning Management System (LMS) and its built-in LA Features

A LMS is a digital platform designed to support the delivery, management, and assessment of learning activities in educational settings. LMSs facilitate the organization of course materials, communication between instructors and students, and the tracking of learner engagement and performance. By mediating a large portion of instructional activities, LMSs generate detailed interaction data that form a primary data source for Learning Analytics and data-driven decision-making in higher education [55, 56]. In Canada, D2L Brightspace has gained widespread adoption, with 12 out of 19 universities using it as their primary LMS, while other institutions employ alternative platforms, including Moodle and Canvas, as identified in the institutional survey presented in section 3.3 of this thesis. Accordingly, this thesis focuses on D2L Brightspace to examine its existing Learning Analytics capabilities, as the platform is both the most widely adopted LMS in Canadian higher education and the current institutional LMS at the University of Ottawa, enabling the reuse and facilitation of existing LMS infrastructure and data sources for practical LA implementation.

At the University of Ottawa, following the successful migration from Blackboard to Brightspace in 2017, institutional efforts have increasingly focused on transitioning from traditional face-to-face learning to technology-assisted learning. This has led to the design of numerous hybrid

learning courses in 2016 [57] and an increase in the number of 100% online programs at the moment. Also, the university has begun exploring more advanced Brightspace features, including Brightspace Performance+ for analytics to access engagement data [57] This section examines the D2L Brightspace Performance+ LA Dashboard, focusing on S3 for detecting at-risk students and providing interventions. Through this analysis, we identify the Brightspace dashboard's limitations when applied to our data landscape. This leads us to propose a new, tailored LA architecture system in the following section.

According to Brightspace [58], S3 is an Early Intervention System that utilizes predictive analytics to enhance student success, retention, and graduation rates. It measures student performance from the early weeks of the semester and offers educators early indicators and predictions of student success and risk levels. S3 generates final grade predictions using machine learning algorithms applied to historical course data, and these models are adaptable to each course's instructional approach and expectations. Weekly predictions are provided to students in the form of a success index. The success index provides the overall predicted outcome that is primarily based on five engagement domains: Course Access, Content Access, Social Learning, Grades, and Preparedness. S3 offers interactive visualizations that illustrate student performance relative to course expectations and peers. For instance, the risk quadrant dashboard positions students based on their success index and current grade, identifying four risk levels with dots on the chart, as shown in Fig. 4.3. Quadrants are as follows: The withdrawal/ dropout risk (1) is where the student seems to be struggling in terms of both engagement and performance. In this case, the student may be at risk of dropping out. The academic performance risk (2) is where the student seems to be engaged but is struggling in terms of performance. The under-engagement risk (3) is where the student seems not to be engaged yet is achieving high grades. In this case, the student may be under-challenged. The on-track, not at-risk, quadrant (4) is where the student is engaged and achieving high grades. (1) is the student who is struggling in both Academics and Engagement; Quadrant (2) is the student who seems to be engaged but is struggling in terms of academic performance; Quadrant (3) is the student who seems not to be engaged, yet is achieving high grades, so the study seems to be under-challenged with that student. Based on our assessment of the current university data landscape, it was determined that Brightspace S3 was inadequate for the University Of Ottawa due to the following reasons:

- Data Source Limitation: S3 primarily relies on data from the LMS with engagement data. This overlooks valuable student information stored in other university systems, such as demographic data or admission information. This limited data pool hinders the system's ability to create a comprehensive picture of each student's academic journey.
- Course-Level Focus: S3 is primarily designed to track student performance within individual online courses. However, for effective early intervention, an overview of all courses a student is enrolled in at a given time is crucial. This holistic perspective allows for a more comprehensive understanding of a student's academic performance and potential areas of difficulty.

Figure 4.3: The risk-level quadrant dashboard positions students based on their success index and current grades [58]

- Hindered Future Research: The "black-box" nature of S3's data prediction process poses a challenge for future research endeavors. Limited access to underlying data and analysis hinders transparency and restricts researchers' ability to explore further or refine predictive models, ultimately impacting the system's long-term development and improvement

By addressing these limitations through a robust data framework encompassing diverse data sources and a transparent data analysis process, we can establish a more effective LA system for identifying and supporting at-risk engineering students at our university.

4.4 University Datasource landscapes & An Approach of Multidimensional Analysis

This section examines the different data sources available at The University Of Ottawa and the research approach chosen to utilize them.

4.4.1 The student-related data landscape

The student-related data landscape (Fig. 4.4) includes crucial information stored in the uOttawa SIS, such as demographics, historical academic experience, program entry qualifications, and student learning course paths. Additionally, engagement data from the Brightspace LMS, including assignment attempts, quizzes, discussions, content access frequency, and assignment grades, is considered. However, Zoom and Microsoft Teams video meeting data are excluded due to time constraints and data format/processing difficulties. The plan is to collect as much data

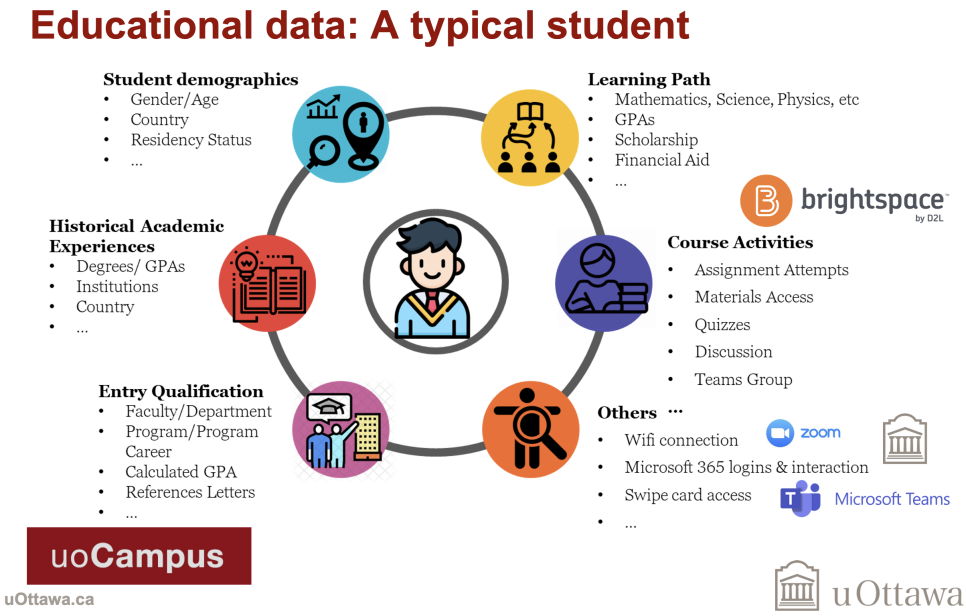


Figure 4.4: Student-Related Data Source lanscape. Source: authors.

as possible within technical limitations. This process will involve multiple phases to explore and understand all data structures and meanings, ensuring preprocessing (including data cleaning and transformation) to transform the data from a database technical context to a more suitable learning context.

At the time of this exploration, a comprehensive list of potential student-related data features was requested for use in Phase 3 (Section 2.2.3). In addition, given the complexity of the Brightspace LMS platform, this study examined and identified relevant data source types and specific datasets, as categorized by Brightspace, that would be required for subsequent analysis. However, due to technical constraints and privacy controls enforced by the University of Ottawa Data Hub team, only a limited subset of the requested data could be made available. The com-

plete list of requested student data features and Brightspace datasets, along with their availability status, is provided in Appendix A, table A.1 and A.2.

4.4.2 Multidimensional Data Model for Analysis

To facilitate a comprehensive analysis of student performance and identify potential risk factors, a multidimensional data model was employed. This model establishes meaningful relationships between various aspects of student data, termed "dimensions." For instance, a university wants to analyze student pathways within their engineering program and identify potential risk factors for dropping out analyzed by dimensions such as student information, program information, academic year, and course information. Each dimension corresponds to a dedicated dimensional table within the system. These tables hold detailed information pertaining to the hierarchical structure of each dimension. Data points, such as grades or program enrollment activities, are categorized as fact data within the data warehouse. This fact data is then aggregated by the various dimensions within the data cube (Fig. 4.5). These implementation steps are as follows:

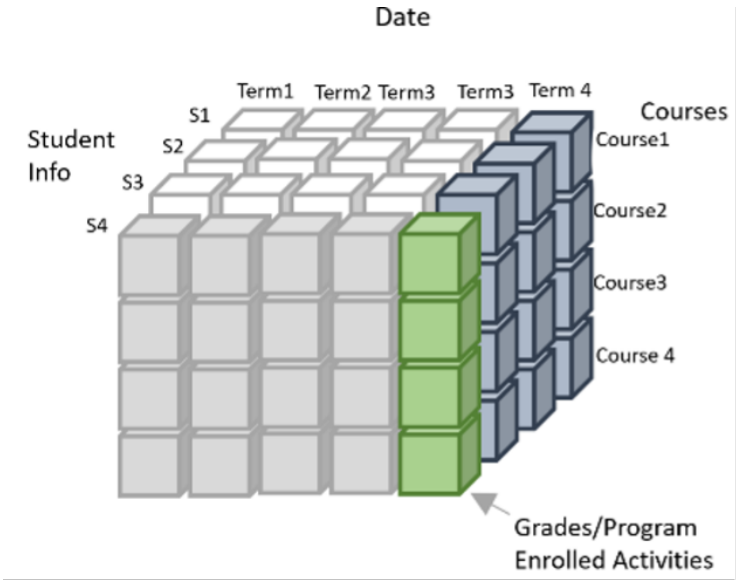


Figure 4.5: A sample of a multidimensional data cube in datawarehouse for student course path analysis Source: authors.

1. Data Preparation: Clean and preprocess data, handle missing values, convert formats, and create new variables if needed.

2. Dimensional Table Creation: Develop tables containing detailed attributes and hierarchies for student, program, course, and semester/year information.

3. Fact Table Creation: Create a fact table with key data points, such as student ID, course ID, semester/year, and grade earned

4. Data Cube Formalization: Based on the focus of analysis, which is at-risk students, define the data cube with related information dimensions like student, program, semester/Year, and Course, linked to the fact table.

5. Analysis: Utilize the data cube to visually represent student course sequences across various dimensions (as depicted in Fig. 4.6). Additionally, employ a range of data mining and machine learning techniques (this step occurred in layer 3 of our system architecture in section 4.3) to discern patterns in successful and unsuccessful pathways.

6. Identify Risk Factors and display them through Visualization or Predictive Models: identify risk factors for student dropout, such as consistently low grades or changing program specializations, using the results from the previous step. These factors are visualized as reports or dashboards occurred in layer 4 of our system architecture in section 4.3.

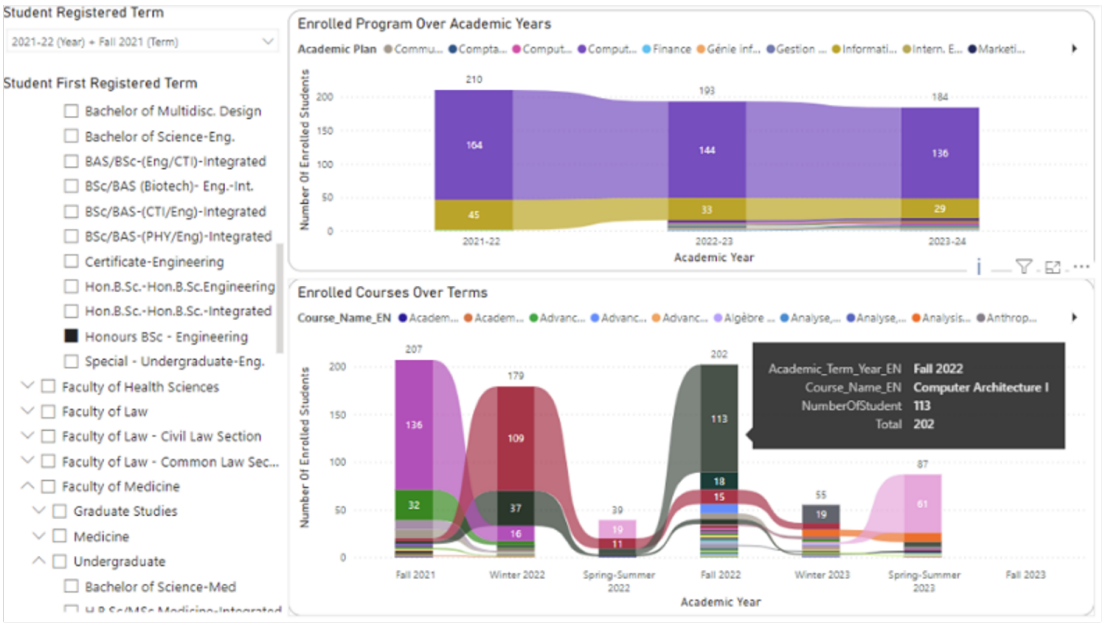


Figure 4.6: A course path of students in the Bachelor of Engineering program are filtered and displayed across various dimensions, including program, term, courses, and students. Source: authors.

Steps 1 through 4 involve transforming the existing Entity-Relationship (E/R) model, currently used in the university databases, into a dimensional model. This transformation leverages the insights and best practices outlined in the reference [16]. This approach allows for a comprehensive analysis of student performance by considering various factors and their interrelationships.

4.5 LA System Architecture Design at Initial Stage

While LA is gaining traction in higher education, the predominant focus lies in predicting student attrition, encompassing both dropout and desertion [10, p. 13] In the initial implementation of LA at the University Of Ottawa, the objective is to establish a data hub for integrating various student data sources and to develop an early warning system for identifying at-risk students based on this integrated data. To address the limitations of Brightspace (as discussed in part 4.1), a robust system architecture was designed, as illustrated in Fig. 4.7, to enhance the accuracy of the at-risk student predictions. This section provides detailed insights into this data model and its advantages. The proposed system comprises four layers for data processing and analysis.

The following layered architecture approach prioritizes early identification of at-risk students. By integrating diverse data sources (layer 1), the system provides a more comprehensive picture of student performance and the factors contributing to risk. Layers 2, 3, and 4 further facilitate data sharing and analysis across different stakeholders and future research endeavors.

4.5.1 Layer 1- Data Sources

This layer encodes all potential educational data sources before duplicating and storing them in the next layer. The data used for the early warning system includes student personal data, such as demographic or admission background, as well as their academic activities in the LMS, including course information, activities, and engagement information. Other sources, such as online learning meeting activities in tools like Microsoft Teams or Zoom, will also be considered in the next phase.

The primary benefit of this layer is the preservation of data provenance and auditability, enabling reproducibility and reprocessing when source systems or extraction logic change. By isolating raw data, the architecture minimizes the risk of data loss or unintended transformation effects in downstream processes.

From a collaboration perspective, this layer allows data engineers to focus on data ingestion and extraction pipelines, while data administrators manage access control, data governance, and compliance. Downstream users are insulated from source-level system complexity.

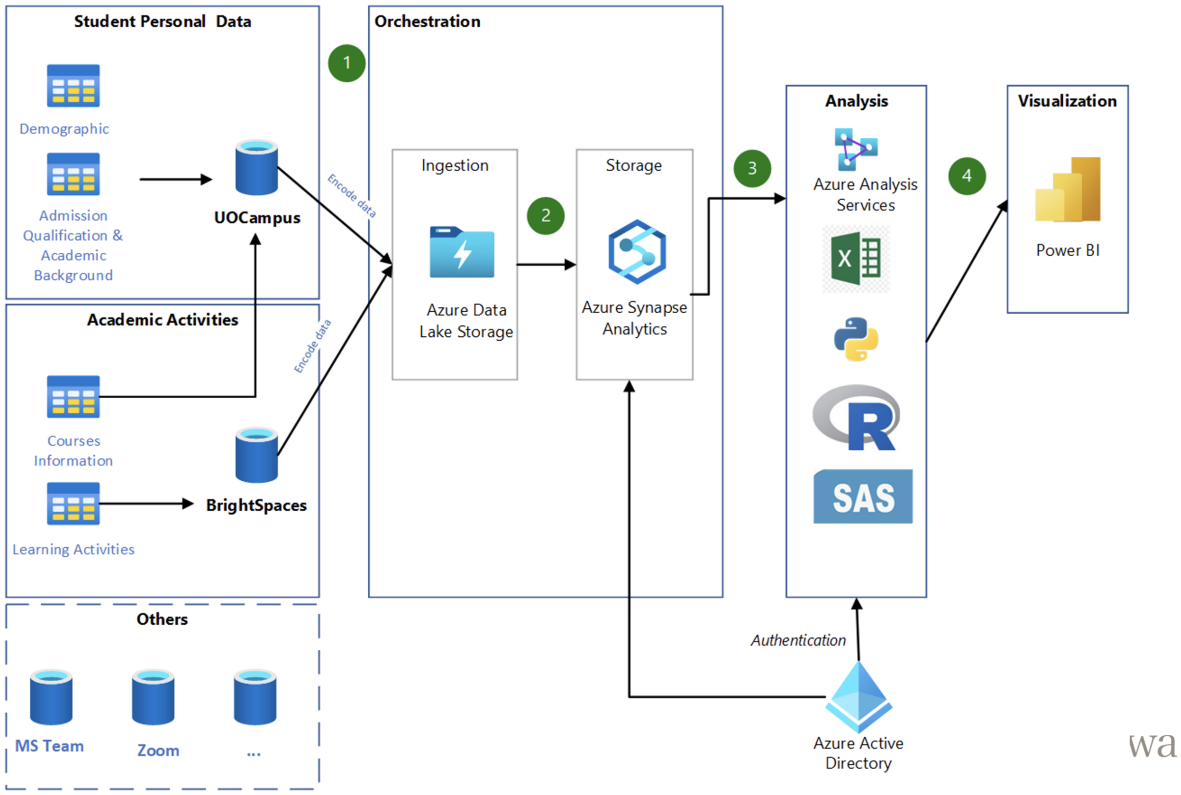


Figure 4.7: Proposed LA system architecture in the initial stage. Source: authors.

4.5.2 Layer 2- Data Orchestration

This layer plays a critical role in data preparation and management. It consists of two key functions:

- Data Ingestion: This sub-layer focuses on acquiring data from diverse sources in layer 1 (e.g., student demographics, LMS activity logs). The data undergoes cleaning, transformation, and loading processes to ensure consistency and quality before storage
- Data Storage: This sub-layer acts as the university's central data hub, leveraging a multi-dimensional data model to optimize query performance for analyzing student data across various dimensions (see section 4.2). This optimized structure allows for efficient retrieval and analysis when exploring student performance across different facets.

Various database technologies are considered to achieve this functionality. The choice of specific tools depends on the chosen data platform. Potential options include cloud-based solutions like

Azure Data lake Storage for raw data storage and Azure Synapse Analytics for data transformation and analysis.

This layer improves data quality and consistency by centralizing transformation logic and business rules. It decouples raw data ingestion from analytical usage, enabling reliable and comparable analyses across cohorts and time periods.

In collaborative settings, data engineers implement ETL/ELT workflows and pipeline automation, while data administrators enforce schema standards, metadata management, and version control. Data analysts and data scientists benefit from access to clean, structured data without duplicating preprocessing efforts.

4.5.3 Layer 3- Data Mining & Data Analysis

This layer unlocks the potential of the data hub for insights and exploration. It utilizes a spectrum of data analysis techniques, ranging from traditional methods like Microsoft Excel to advanced tools like machine learning and deep learning algorithms. By leveraging the data in the hub (layer 2), this layer empowers stakeholders to perform in-depth analysis and data mining, uncovering valuable patterns and trends related to student performance. Additionally, controlled access to this layer is granted through secure authentication tools like Azure Active Directory. This not only safeguards sensitive student data but also facilitates collaboration and knowledge sharing among authorized stakeholders, further enhancing the potential for future research endeavors utilizing the university's data hub.

The separation of analytical data from upstream layers enables flexible experimentation and iterative model development without compromising data integrity. This layer supports reproducible analyses, feature reuse, and parallel evaluation of multiple modeling approaches.

From a collaboration standpoint, data scientists focus on feature engineering, model development, and evaluation, while data analysts perform exploratory and descriptive analyses. This separation reduces coupling between analytical experimentation and core data pipelines.

4.5.4 Layer 4: Visualization & Reporting

This layer focuses on transforming insights and results from layer 3 into understandable formats for effective communication. It utilizes visualization tools like Power BI to translate complex data patterns and findings into clear and compelling visuals, such as charts, graphs, and dashboards. These visualizations facilitate knowledge sharing and communication with various stakeholders within the university, empowering them to gain valuable insights into student performance and make informed decisions.

This layer ensures that complex analytical outputs are translated into interpretable insights for non-technical users. By separating visualization from computation, the architecture improves system performance, usability, and data governance.

In collaborative environments, data analysts design dashboards and key indicators, while data administrators control data exposure and privacy policies. Decision makers can access actionable insights without interacting directly with raw or analytical data layers.

4.6 Summary and Discussion

This chapter examined existing LA system architectures and institutional implementation practices to address Research Question 2. Through a review of prior literature and an exploration of available data sources at the University of Ottawa, several key challenges were identified, including fragmented data sources, limited interoperability that requires close collaboration with the Data Hub team, and governance and privacy constraints that restrict access to requested data. The analysis of Brightspace's S3 indicates that LMS-provided analytics support basic monitoring but offer limited flexibility for institution-specific risk definitions and advanced analytical use cases. At uOttawa, reliance on LMS-level tools such as Brightspace constrains customization, reuse of analytical components, and longitudinal analysis. These explorations highlight the need for an integrated, scalable, and context-aware LA architecture, which directly motivates the system design proposed in this chapter. The following chapter 6 apply and implement this design to demonstrate its practical feasibility.

Chapter 5

A Systematic Review: Using Learning Analytics and Machine Learning for Early Detection of At-risk Students in Higher Educational Institutions.

In this section, We conducted a systematic literature review focusing on the evaluation of ML prediction models using LA data in HEI. The review emphasizes empirical data from LMS and SIS, excluding self-reported datasets. To ensure relevance, only studies published from 2019 onward were considered. A total of 23 relevant articles were retrieved from Scopus and IEEE Xplore to address the key research questions:(1) What ML tasks are most commonly used for predicting at-risk students and their scope of applicability?(2) What evaluation methodologies are applied to assess the effectiveness of ML models in predicting student performance?

5.1 Motivation

In recent years, the application of machine learning models for predicting at-risk student performance has garnered significant attention within the realm of Learning Analytics research. This burgeoning field holds the promise of providing timely interventions to support struggling students, ultimately enhancing their chances of academic success. However, amidst this burgeoning interest, it is imperative to recognize the pivotal role that methodologies play in the assessment and validation of these predictive models.

The evaluation of machine learning models designed for at-risk student prediction necessitates a judicious consideration of diverse methodological approaches. As researchers seek to refine and advance these models, the choice of methodology becomes a critical determinant of the model's reliability, generalizability, and applicability in real-world educational settings. This literature review embarks on a comprehensive exploration of the various methodologies employed in the assessment of machine learning models for at-risk student prediction

5.2 Previous Survey Works

We've identified recent peer-reviewed literature surveys (since 2021) on predicting student performance through Learning Analytics & Machine Learning. These reviews offer diverse approaches and valuable insights. Albreiki et al. [59] listed the data features and the machine learning algorithms employed across the corpus of literature through four distinct lenses: Predicting at-risk student performance, Determining students' dropout rates, Evaluating Students' Performance based on Static and Dynamic Data, and Formulating Remedial Action Plans. They systematically reviewed 78 worldwide studies spanning from 2009 to 2021. The results of this review highlighted the wide array of Machine Learning (ML) techniques employed to address the challenges related to predicting at-risk students and anticipating student dropouts. Furthermore, it was observed that most studies utilized two types of datasets: data sourced from student colleges/university databases and information from online learning platforms.

Shafiq et al. [29] conducted a comprehensive survey encompassing 100 articles published between 2017 and 2021, leading to the development of an extensive taxonomy of Machine Learning methodologies (Figure 4). Furthermore, the reviewers delineated pertinent factors associated with both successful and unsuccessful student outcomes, organizing them into distinct categories and sub-categories (Figure 5). Additionally, the study elucidated several identified limitations in the extant research, including non-generalizability of results, issues with data imbalances and overfitting, an overarching reliance on a student-centric approach, and constraints imposed by limited sample sizes. Moreover, the researchers underscored a critical lapse in addressing ethical considerations related to student privacy data. In terms of machine learning methods, the reviewers stressed the importance of unsupervised learning approaches in understanding student behavior, an area that has received less emphasis in existing literature. This observation is reiterated in the concluding remarks of several literature reviews [29], [59], [60].

Cano and Leonard [37] undertook an exhaustive categorization of data feature sets utilized in 60 scholarly works spanning the years 2006 to 2021. They classified these features into five distinct categories: Academic Performance; Demographic & Academic Performances; Demographic & Academic Performances & Family Background; Soft Skill & Extra-curricular; and Learning Management Systems (LMS) Activities; as well as SoftSkills. Additionally, they underscored the significance of feature selection methodologies in the optimization of Machine Learning

Models, enumerating numerous techniques including Genetic Algorithms (GA), Artificial Neural Networks (ANN), k-Nearest Neighbors (KNN), Fuzzy Technique for Order of Preference by Similarity to Ideal Solution (Fuzzy TOPSIS), various feature selection (FS) algorithms, filter-based algorithms, Particle Swarm Optimization (PSO), Support Vector Machines (SVM), and Random Forests (RF). Noteworthy is the prevalence of researchers employing hybrid methodologies in neural networks, specifically the integration of Fuzzy techniques and Feature Selection algorithms, such as FUZZY-Artificial Neural Network (ANN) and FUZZY-TOPSIS. Cano and Leonard [37] similarly adopts a concentrated emphasis on feature selection and the utilization of neural networks in their review. In a subsequent review conducted in 2023, they expound upon prevalent feature selection techniques, encompassing the Fisher Score, Mean Absolute Difference (MAD), Variance Threshold, and Correlation Threshold. Additionally, the review underscores two common approaches: the hybrid methodology, denoted as "ensemble learning" methodology, and the utilization of Artificial Neural Networks (ANN) in conjunction with FUZZY-ARTMAP, a method employed to mitigate the limitations associated with standalone ANN models.

To summarize, recent surveys have homed in on two primary areas of focus. The first centers on the application of Machine Learning (ML) Techniques and Dataset Features in training data, while the second delves into Neural Network techniques and its Feature Selection methods to enhance the model performances. Interestingly, there appears to be a relative scarcity of emphasis on the evaluation methodologies and empirical testing to validate the effectiveness of ML applications in early prediction. Additionally, in other techniques than Neuro Network, aspects like data exploration or pre-processing and Feature engineering, have received comparatively less attention, potentially exerting a substantial influence on ML performance.

5.3 Scope of Discussion and Objectives

This systematic literature review seeks to bridge a research gap by dedicating specific attention to the crucial areas of Evaluation for ML prediction models using LA data features. It is important to note that the discussion is delimited to the domain of higher education environments, with a specific focus on empirical databases sourced directly imported from related institutional sources, rather than synthesis data by self-report methods (such as Excel, surveys, words files, etc.). Finally, to ensure the inclusion of the most current and state-of-the-art literature, this review will exclusively consider studies published from 2019 onwards for collection and analysis.

5.4 Literature Review Methodology

5.4.1 Research Questions

The literature review has been carried out to provide elaborate answers to the following key research questions formulated below:

- RQ1: What ML tasks are most commonly used for predicting at-risk students and their scope of applicability?
- RQ2 What evaluation methodologies are applied to assess the effectiveness of ML models in predicting student performances?

Add justification.

5.4.2 Data Sources

In order to carry out an extensive systematic literature review based on the objectives of this review, we exploited two research databases to find the primary data and to search for the relevant papers. The databases consulted in the entire research process are provided in Table 5.1. We searched the above search string in two major digital libraries, which are IEEE Xplore, Scopus. The search string is modified and translated to the proper input query for searching each digital library. We only focused on peer-reviewed journals and conference articles and excluded the book chapters and other types of publications. The search was conducted on September 24th, 2023. Table 1 shows the search results and filters used by each database.

Table 5.1: Database Search Summary

Database	Website	Search In	Results	Filter Used
Scopus	https://www.scopus.com/	Abstract	63	Subject Area = Computer Science & Engineering & Decision Science; Document Type = Articles & Conference Papers
IEEE	http://ieeexplore.ieee.org/	All	107	None

5.4.3 Search Query

The query consists of three main components. The first part captures “Machine Learning” related terms, and the second one captures “Higher Education” associated terms, the last captures “At-risk students prediction” terminology: (”Learning Analytics” OR ”E-Learning” OR ”Learning Management system” OR ”Educational Data Mining”)

AND

(”Machine Learning” OR ”Artificial Intelligence” OR ”AI”) AND (”Higher Education” OR ”University” OR ”College” OR ”Post-Secondary”)

AND

(”Student Performance” OR ”Student Improvement” OR ”AT-RISK Student” OR ”Drop-out” OR ”Retention”) AND (”Detection” OR ”Prediction” OR ”Monitor”)

5.4.4 Inclusion, Exclusion Criteria and Literature Selection

The retrieved articles from the digital libraries were excluded based on the exclusion criteria. Table 5.2 presents both inclusion and exclusion criteria.

Table 5.2: Inclusion and Exclusion Criteria for the Systematic Review

Inclusion Criteria	Exclusion Criteria
I1: Papers conducted in the context of higher education using higher education data.	**E1:** Papers not published in peer-reviewed journals or conferences.
I2: Papers that performed adequate testing and evaluation processes to demonstrate actual results and effects of Machine Learning (ML) or Artificial Intelligence (AI).	**E2:** Papers using datasets derived solely from self-reported sources (e.g., interviews or researcher-administered surveys).
I3: Papers that describe methodologies aimed at enhancing predictive models and substantiating their superiority through adequate evaluation.	**E3:** Papers published in languages other than English.
	E4: Papers conducted before 2019.
	E5: Papers that do not satisfy both I2 and I3.

Selection Steps- The paper selection steps are represented as below:

- Remove not peer-review articles/conferences: 110 articles. Remains : 60

- Duplicated or not available in full text: 5 articles. Remains: 55
- Remove research conducted before 2019:4 articles. Remains 51
- Abstract screening: Articles abstracts are screened out based on the inclusion and exclusion criteria, 45 studies remained to be screened.
- Full-text screening: From the 43 articles that were left, we excluded studies according to the exclusion/inclusive, 24 studies remain.

5.5 Background: Evaluation Metrics for Predictive Modeling

The evaluation of machine learning models is a critical step in assessing their effectiveness and reliability. Different evaluation metrics capture distinct aspects of model performance, each with specific strengths and limitations. In the context of LA, student performance prediction often involves imbalanced datasets, limited sample sizes, and a high risk of overfitting [29, 59]. Consequently, careful selection and interpretation of evaluation metrics is essential to ensure meaningful and reliable assessment. This section introduces the most commonly used evaluation metrics in ML–based LA studies.

- **Precision:** Precision measures the proportion of correctly predicted positive instances among all instances predicted as positive. A high precision value indicates a low false-positive rate.
$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.1}$$
where TP denotes True Positives and FP denotes False Positives.
- **Recall:** Recall, also known as sensitivity, represents the proportion of correctly predicted positive instances relative to all actual positive instances.
$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.2}$$
where FN denotes False Negatives.
- **F1-score:** The F1-score is the harmonic mean of Precision and Recall, balancing false positives and false negatives. It is particularly useful when dealing with imbalanced class distributions.
$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.3}$$
When the costs of false positives and false negatives differ, the F1-score provides a more informative measure than Accuracy alone [**lit12**].

- **Area Under the Curve (AUC):** AUC represents the area under the Receiver Operating Characteristic (ROC) curve and evaluates a model's ability to distinguish between classes across varying decision thresholds. A higher AUC indicates stronger discriminatory capability, such as distinguishing between students who pass or fail a course.

- **Accuracy:** Accuracy measures the proportion of correctly classified instances among all instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{5.4}$$

  where TN denotes True Negatives. Accuracy is most suitable when class distributions are balanced and misclassification costs are similar.

- **Specificity:** Specificity evaluates the proportion of correctly identified negative instances. In the context of student performance prediction, it reflects the model's ability to correctly identify students who fail.

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{5.5}$$

- **Matthews Correlation Coefficient (MCC):** MCC measures the correlation between predicted and actual classifications and provides a balanced evaluation even for imbalanced datasets.

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{5.6}$$

- **Geometric Mean (GM):** The geometric mean is particularly useful for imbalanced datasets, as it balances performance across classes. However, in cases of overfitting, GM may overestimate true model performance.

- **Cohen's Kappa:** Kappa measures agreement between predicted and actual classifications while accounting for agreement occurring by chance. Although informative, Kappa may be less reliable in heavily overfitted models.

- **Mean Absolute Error (MAE):** MAE quantifies the average magnitude of prediction errors by computing the mean of absolute differences between predicted and actual values.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{5.7}$$

  Lower MAE values indicate predictions closer to actual student performance outcomes.

- **Root Mean Square Error (RMSE):** RMSE measures the square root of the average squared differences between predicted and actual values, placing greater emphasis on larger

errors.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5.8}$$

RMSE is particularly sensitive to large prediction errors and is commonly used in regression-based student performance prediction tasks.

5.6 Results and Key Findings

5.6.1 What ML tasks are most commonly used for predicting at-risk students and their scope of applicability?

To better understand model applicability, we categorized each study's scope into three level: (1) Institution-level: Models applied broadly across multiple departments or programs within a HEI;(2) Program-level: Models focused on students within a specific academic program; (3) Course-level: Models developed for a particular course, typically using data from one or a few course offerings (Table 5.3)

The literature in Table 5.3 reveals that most studies focus on binary classification tasks to predict student success or risk of failure at the course level, where data is more accessible and outcomes are clearly defined. Fewer studies have explored multi-class classification for more nuanced outcomes or regression for continuous predictions like Grade Point Average (GPA). Moreover, the majority of models are scoped at the course level (57%), with fewer applied at the program (17.4%) or institutional level (26%), suggesting a need for broader, more scalable modelling approaches that generalize across academic contexts.

5.6.2 What evaluation methodologies are applied to assess the effectiveness of ML models in predicting student performances?

Evaluating classification models typically involves diverse performance metrics, each highlighting different aspects of model effectiveness. Figure 5.1 summarizes all evaluation methods used with best performance algorithm.

Among the 23 reviewed studies, about 91% applied classification tasks—most commonly using Accuracy, Recall, Precision, and F1-score (Section 5.6.2.1)—while only two employed regression methods (Section 5.6.2.2). Beyond standard performance metrics, statistical tests enhance model evaluation by offering a formal, unbiased way to assess differences between algorithms. They

Table 5.3: Literature Review: Summarizes Selected Studies

ID-Study	ML Task	Prediction Objectives	Model Applicability Scope	Source Types	Data Scopes (Filters)			
					No. of Students	No. of Courses	No. of Acad. Years	No. of Records
1-[61]	Regression	Assignments' Scores	Program	Public (OULAD)	1627	-	-	10354
2-[62]	Binary Class.	Course Pass/Fail	Institution	LMS (Moodle)	-	-	-	54803
3-[63]	Binary Class.	Course Pass/Fail	Course	Public (OULAD)	-	7	2	-
4-[64]	Binary Class.	Course Pass/Fail	Course	Public (Kaggle)	-	7	1	-
5-[65]	Binary Class.	Course Pass/Fail	Course	LMS (Canvas)	-	1	2	-
6-[66]	Binary Class.	University Drop Out	Institution	LMS & SIS	-	-	-	-
7-[67]	Multi-Class.	Final CGPA	Institution	LMS & SIS	226	-	5	-
8-[68]	Binary Class.	Course Pass/Fail	Course	LMS	-	1	1	-
9-[69]	Binary Class.	Course Pass/Fail	Institution	LMS	-	-	23	-
11-[70]	Binary Class.	Course Pass/Fail	Course	LMS (Moodle)	105	1	1	7057
12-[71]	Binary Class.	GPA Levels	Course	Public (Kaggle)	-	2	3	649
13-[60]	Binary Class.	Course Pass/Fail	Course	LMS (Moodle)	1523	1	1	-
14-[37]	Binary Class.	Course Pass/Fail; Course Dropout	Institution	SIS	-	-	1	-
15-[72]	Multi-Class.	Course Grade Levels	Institution	LMS	3569	-	2	1567481
22-[73]	Binary Class.	Probation Year Pass/Fail	Program	SIS	-	-	4	1761
24-[74]	Multi-Class.	Course Results(Excelent, good, average, poor)	Course	SIS	140	1	1/3	-
25-[75]	Binary Class.	Pass/Fail	Course	SIS	3225	1	2	-
26-[76]	Multi-Class.	Course Grades (high, normal, low)	Course	LMS & SIS	50	1	-	-
28-[77]	Multi-Class.	First year CGPA (Excellent, Very good, Good, Aceptable, Poor)	Program	LMS	1430	-	2	-
29-[78]	Multi-Class.	Course Results (withdraw, fail, pass, distinction)	Course	Public (OULAD)	32593	7	-	-
30-[79]	Multi-Class.	Course Grade (A,B,C)	Program	LMS & SIS	-	43	-	-
31-[80]	Regression	GPA Scores	Course	Public (StudentLife)	48	1	1/3	-
32-[81]	Multi-Class.	CourseGrade (A,B,C,D)	Course	SIS	600	10	-	-
This study	Binary Class.	Course Pass/Fail	Program	LMS & SIS	2591	339	1/3	3496

Table Abbreviations: Class.: Classification, CGPA:Cumulative Grade Point Average , GPA:Grade Point Average, OULAD: Open University LA Dataset

		Classification Task								Regression Task			Statistical Test				Emp		Studies'
ID	**Best Algorithms**	Acc	Rec	Pre	F1	Spec	G-Mean	Auc/Roc	MCC	RMSE	MAE	Adj-R^2	Wil.	Bon.	T	Kap.	**Test**	**Cnt**	**Group**
2	SVM, LR [4]	✓	✓	✓	✓													**4**	A
7	RF[2]	✓	✓	✓	✓													**4**	A
11	RF[2]	✓	✓	✓	✓													**4**	A
12	CNN[3]	✓	✓	✓	✓													**4**	A
21	MA-DL[3]	✓	✓	✓	✓													**4**	A
26	DT(J48)[2]	✓	✓	✓	✓										✓			**5**	A
28	ANN[3]	✓	✓	✓	✓												✓	**5**	A
29	RF[2] & GBM[1]	✓	✓	✓	✓													**4**	A
32	XGB[1]	✓	✓	✓	✓													**4**	A
4	GBM[1]	✓	✓	✓	✓			✓										**5**	B
5	CatBoost[1]	✓	✓	✓	✓			✓										**5**	B
6	RF[2]	✓	✓	✓	✓			✓										**5**	B
13	RF[2]	✓	✓	✓	✓	✓			✓									**6**	B
9	ANN[3]	✓																**1**	C
22	SVM, LR[4]		✓															**1**	C
24	DT[2]	✓																**1**	C
30	CART[2]	✓															✓	**2**	C
3	ANN MIL[3]	✓	✓			✓							✓					**4**	D
8	DGN TM[3]							✓										**1**	D
14	MVGP[3]	✓	✓			✓	✓	✓					✓	✓		✓		**8**	D
25	GLMNET[4]							✓										**1**	D
1	SVR [4]										✓							**1**	RG
31	LassoCV									✓	✓	✓						**3**	RG
	No. Of. Studies	18	16	13	13	3	1	6	1	1	2	1	2	1	1	1	2		
	Percentage	78%	70%	57%	57%	13%	4%	26%	4%	4%	9%	4%	9%	4%	4%	4%	9%		

Columns Abbreviations: Acc: Accuracy, Rec: Recall, Pre: Precision, F1: F1-Measure, Spec: Specificity, G-Mean: Geometric Mean, Auc/Roc: Area Under the Receiver Operating Characteristic curve, MCC: Matthews Correlation Coefficient, RMSE: Root Mean Square Error, MAE: Mean Absolute Error or Mean Absolute Percentage Error, Adj-R^2: Adjusted R-squared, Wil.: Wilcoxon rank-sum test, Bon,: Bonferroni–Dunn test, Kap.: Kappa scores, Emp.: Empirical Testing, Cnt: Total number of evaluation metrics used in a study. [1]**Boosting Algorithms**: GBM: Gradient Boosting Machine, XGB: Extreme Gradient Boosting Machine, CatBoost [2]**Tree-based Algorithms**:RF: Random Forest, DT: Decision Tree, CART: Classification and Regression Trees [3]**Neural Networks**:CNN: Convolutional Neural Network, ANN: Artificial Neural Networks, MIL: Multiple Instance Learning Approach, MVGP: Multi-view Genetic Programming, MA-DL: Multi-Attention Deep Learning, DGN TM: Deep Learning GridNet with TIMEMASK structures [4]**Others**:SVR: Support Vector Regression, SVM: Support Vector Machine, LR: Logistic Regression, GLMNET: Generalized Linear Model with Elastic Net

Figure 5.1: Literature Review: ML Model Performance Metrics and Evaluation Techniques in Reviewed Studies

support more reliable conclusions and reduce the risk of errors from small samples or subjective judgment. However, in our review, only about 12.5% of studies applied such tests—specifically the t-test, Wilcoxon signed-rank, and Bonferroni-Dunn—after reporting performance metrics (Section 5.6.2.3), and just 8.6% conducted empirical testing with real-world data to validate the model generalization (Section 5.6.2.4). Next sections provides details of how these evaluation techniques are applied and the studies' model results.

5.6.2.1 Evaluation Metrics For Classification ML Task (Studies in Group A,B,C,D)

Group A: Studies Use Accuracy, Precision, Recall, F1-Score (39%) Turabieh [71] conducted two experiments to compare prediction algorithms, one with the inclusion of feature selection and another without. In both experiments, all four metrics—Accuracy (0.93-0.95), Precision (0.88-0.92), Recall (0.89-0.91), and F1 Measure (0.9-0.91)—exhibited higher values than those of

the other algorithms, with improvements ranging from approximately 7% to around 24% across the board. Consequently, the author concluded "Convolutional neural network (CNN) approach outperforms other classifiers based on Accuracy value". Additionally, Boxplot diagrams based on Accuracy values were visualized for each experiment, providing further support for this conclusion. In a similar vein, Chen and Zhai [81] conducted an evaluation wherein they determined that the XGB technique demonstrated superior performance across four key performance metrics. Specifically, the achieved Accuracy rate was recorded at 83.56%, Precision rate at 84.30%, Recall rate at 79.66%, and F-measure rate at 81.91%. To substantiate this conclusion, the authors supplemented their findings with a chart representation of the performance metrics stratified by each utilized algorithm. Bai et al. [72] proposed a student performance prediction model employing a MA-DL (Multi-attention Deep Learning), categorizing students' average scores into three levels: A, B, and C, wherein the top 20% of students fall into class A, the bottom 60% into class B, and the remaining 20% into class C. The study conducted a comparative analysis of their model against conventional approaches such as Linear Regression (LR), DT, SVM, and RF. Results indicated that their model surpassed the traditional methods in all four performance metrics. Notably, the Accuracy achieved on the dataset was 82.25%, exhibiting a 13.84% improvement over the best-performing traditional method. In a more specialized investigation, Gaftandzhieva et al. [70] conducted four experiments using distinct datasets to assess their prediction model employing RF, K-Nearest Neighbors (KNN), XGB, and SVM algorithms. The datasets encompassed four-week and eight-week data from a semester-long online course, with variations in data representation involving balanced and unbalanced techniques. The findings indicate that the model performed moderately well, achieving a prediction Accuracy of 78%. Among the algorithms, RF exhibited superior performance, particularly in Recall, where it achieved a value of 72% surpassing SVM. Furthermore, RF demonstrated the highest F1-score at 0.77, whereas SVM recorded the lowest at 0.70. In order to compare the influence of balanced and imbalanced datasets on performance, the authors employed RF results to highlight the enhanced accuracy metrics observed with the balanced dataset, showcasing an improvement of 8 to reach an Accuracy metric of 78%. Furthermore, the study concluded that "RF algorithm may be used to predict which students will fail after eight weeks" as gleaned from a comparative analysis between 4-week and 8-week data. In contrast, certain studies exhibit ambiguity in determining the champion algorithm due to varying performance trends across the four key metrics. In their study investigating the feasibility of predicting applicants' early academic performance at university based on admission criteria, Mengash [77] identified that the Artificial neural network (ANN) classifier demonstrated superior performance in two pivotal metrics: an Accuracy rate of 79%, reflecting classifier effectiveness, and a Precision rate of 81%, signifying classifier predictive power. Conversely, the DT classifier excelled in both Recall rate (80%), denoting classifier Sensitivity, and F1-Measure rate (81%), indicating a harmonious balance between Recall and Precision. Consequently, the authors concluded that, overall, the ANN classifier technique outperformed the other approaches. Likely, [62], segmented the dataset into different time stages, ranging from 25% to 100% of the course timeline. They conducted experiments and compared various algorithms. DT emerged

as the champion, particularly in the final stage (100% of the course), showcasing high Accuracy, Precision, and F1-measure values, all surpassing 80%. While the Recall value was slightly lower at 79.00%, it remained noteworthy. The author's conclusion was based solely on this final stage, which demonstrated the highest Accuracy value (80%) among all stages and algorithms.

In certain studies, the champion algorithm is determined based on the combined use of Accuracy and the F1 score, which provides a balanced assessment by considering both Precision and Recall in classification tasks. Alhazmi and Sheneamer [67] employed a unique approach to predict student GPA categories, involving rounding GPAs to the nearest decimal point within specified ranges (0.10, 0.30, 0.50) before assigning them to grade categories A, B, C, or D. The authors conducted three distinct experiments based on these rounding methodologies, evaluating the performance of six different classifiers. Subsequently, RF is identified as the top-performing classifiers, primarily based on F1 Score. The study also concluded that the rounding methods significantly enhanced the classifiers in term of Accuracy value, supported by the presentation of data in tabular form encompassing four performance metrics and the inclusion of confusion matrices. Adnan et al. [78] conducted a study aimed at enhancing the performance of DT and Feedforward Neural Network (FFNN) models. While the authors primarily assessed the performance differences using Accuracy and F1 score, they also provided metrics for Recall and Precision without further elaboration or mention. Meanwhile, Khan et al. [76] assessed the performance of various classifiers utilizing data from students' initial exams, which accounted for 15% of the overall grades. Their evaluation led them to conclude that the DT algorithm (specifically, DT J48) exhibited the highest effectiveness, primarily due to its superior Accuracy. Additionally, the authors conducted a t-test (details will be discussed in the section 5.6.2.3 Statistical Testing) based on the F1-score to substantiate their decision.

Group B: Studies Use Accuracy, Precision, Recall, F1-Score with Other Metrics (17%) Anderková et al. [64] divided the initial dataset into five intervals, each corresponding to specific time periods. This segmentation was employed to monitor the progress of individual students over the weeks and to make timely predictions about their likelihood of semester completion, specifically in a Binary Class. scenario (pass or fail). The performance evaluation was carried out using four metrics, comparing the results generated by three distinct classification algorithms: RF, Neural Network (NN), and Gradient Boosting Machine (GBM), across five different course timelines (20%,40%, 60%, 80%, and 100%). Given the notably higher percentages across all four metrics, it can be inferred that the models exhibit greater effectiveness at the 80% course timeline across all algorithms. However, further analysis determined that GBM displayed superior performance, resulting in the fewest misclassified examples for the target "Fail" (i.e., a lower false positive rate), a conclusion substantiated by the ROC curve. Dileep et al. [65] found that the CatBoost algorithm outperforms RF and XGB. This is evidenced by its higher values in Precision, Recall, F1 measure, Accuracy, and Area under the receiver-operating characteristic curve (AUC) across both online and in-person course datasets at week 3, and week 6 accumulated data samples, respectively. Meanwhile, according to Park and Yoo [66], RF emerged as the

top-performing algorithm with an Accuracy of 96%, F-measure of 84%, and AUC of 95%. This performance surpassed that of DT, SVM, and the Deep Neural Network (DNN). Recording to the study, DNN also demonstrated commendable performance, albeit with slightly lower Accuracy at 0.85 and AUC at 0.77, however, it exhibited a notably low F1-Measure at 0.3. It appears that in this assessment, the authors placed greater emphasis on Accuracy and AUC, although they did not explicitly provide a rationale for this prioritization.

In addition to the commonly employed evaluation metrics including AUC and Receiver-operating characteristic curve (ROC), Jokhan et al. [60] incorporate Specificity and Matthews Correlation Coefficient (MCC) as critical evaluation criteria in their study. The authors establish their evaluation framework with the principle that "A best predictor is the one that achieves high performance in the five statistical measures discussed. However, it should perform better at least in some of the measures compared to the existing predictors. A predictor that is unable to predict passed or failed students correctly cannot be used for prediction" Subsequently, the authors partition the dataset into three distinct subsets corresponding to different time points within a course timeline (specifically, at 6 weeks, 8 weeks, and 10 weeks) and assess their predictive efficacy when employed for training a RF classifier. Ultimately, based on these metrics, the classifier trained on data at the 8-week mark is deemed the most effective.

Group C: Studies Exclusively Employ one metric from four common metrics (Accuracy, Recall, Precision, F1 Score) with or without Another (17%)

In certain studies, relying solely on Accuracy as the primary metric for drawing conclusions raises concerns about the comprehensiveness of their evaluation. Ghashout et al. [69] identified the ANN as the primary algorithm for their novel model, the Hierarchical ANN, aimed at enhancing prediction performance. In the initial experiment, the ANN exhibited an Accuracy of approximately 57%, surpassing KNN, SVM, and DT. Subsequently, the Accuracy was enhanced to approximately 71%. Notably, the determination of the champion algorithm was exclusively based on Accuracy and employed a k-fold cross-validation method, with no consideration of additional metrics. Furthermore, the evaluation of their new model's performance, which indicated an improvement in prediction Accuracy, was solely reliant on Accuracy as the assessment criterion. Similarly, Joshi et al. [74] constructed a prediction model, determining that the DT classifier exhibited superior performance, achieving an Accuracy of 96%. In contrast, RF, SVM, Gaussian Naive Bayes (GNB), and XGB achieved Accuracies of 93%, 93%, 89%, and 79% respectively. The authors also demonstrated low bias and low variance in their model by presenting a validation curve depicting the fluctuation of training and validation errors across the training dataset, although they did not provide an explicit rationale.

On the other hand, some researchers, while primarily considering Accuracy, employ additional validation methods. For instance, Khan et al. [79] identified Classification and Regression Trees (CART) as their champion algorithm based on a commendable Accuracy rate of 92.73%. Although they presented other metrics including Precision and Recall, no in-depth analysis or interpretation was provided for these scores. Nevertheless, they meticulously subjected the model

to further testing with a new dataset, yielding bad performance results below 40%. Xiang et al. [61] adopted an approach involving the segmentation of data samples by subcategories defined by four distinct features: gender, parent marital status, health, and family size. The authors assessed the performance of ML models in each subcategory, favouring those demonstrating higher Accuracy values and lower standard deviations, indicative of reduced bias. The study provides detailed Accuracy statistics, including minimum and maximum values, as well as standard deviations, enabling the identification of data features influencing model bias. Meanwhile, Neda et al. [73] emphasized the use of Recall metrics as a priority in evaluating their model, particularly in the context of an imbalanced dataset. In the study, the authors observed a significant impact of imbalanced datasets, particularly in the early stages of training. This prompted the implementation of oversampling techniques approach. While Accuracy initially exceeded 75%, a closer examination revealed low Recall rates (under 0.2). Despite the high Accuracy, this approach detected less than 20% of at-risk students. By shifting their focus to Recall, the authors were able to improve detection rates for at-risk students.

Group D: Studies Use Only ROC/AUC & Studies Use Accuracy & Recall with-/without Other Metrics (17%)

Some researchers rely solely on ROC/AUC as the primary metric for evaluating prediction models, often without providing explicit justification for this choice. For instance, Wan et al. [68] applied this method to evaluate their Deep Learning model across various stages of learning. They observed an increasing trend in ROC/AUC scores over time, enabling them to identify the most effective time stage for early prediction of at-risk students, specifically in the 8th week with an AUC of 89.67%. Similarly, Bertolini et al. [75] employed generalized linear model with elastic net (GLMNET) and determined that as early as week 3 of the semester, robust predictions of successful course performance can be made.

Given the nature of the problem, especially interesting to focus on students who are likely to fail, some studies use Accuracy, Recall, and Specificity. In [63], the author conduct experiments to compare 23 ML algorithms with different representation datasets which are transforming by Multiple Instance Learning (MIL) techniques called simpleMI and MIWrapper. Based on the experiment results, the authors demonstrates that traditional representation (which usually used in traditional ML prediction model) favours higher specificity but sacrifices Recall (Sensitivity) in predicting student performance in specific course. Conversely, flexible representation through WrapperMI leads to improved sensitivity at the cost of reduced specificity. This trade-off results in overall less accurate predictions. Additionally, when employing SimpleMI for flexible representation, a more balanced performance is achieved in both Recall (Sensitivity) and Specificity. Consequently, this representation approach yields the highest or very close to the highest Accuracy. Likely, Cano and Leonard [37], undertook a comprehensive evaluation of their Multi-view Genetic Programming (MVGP), considering several vital metrics such as Specificity, Recall, Kappa, Geometric Means, and AUC/ROC. This thorough assessment was driven by the objective of ensuring the developed algorithms could effectively handle imbalanced data distributions and

provide accurate predictions for both majority and minority classes. Their findings demonstrated that MVGP, along with Multi-view Ensemble Algorithm (MVMI), outperformed other traditional models, particularly in terms of Accuracy and Geometric Mean (G-Mean). Additionally, MVGP exhibited superior performance in AUC compared to MVMI, while both MVGP and MVMI performed similarly for Kappa, thereby establishing MVGP as the top-performing classifier in their experiment.

5.6.2.2 Evaluation Metrics For Regression ML Task (Studies in Group RG)

Among the reviewed studies, only two have utilized ML techniques specifically for regression tasks aimed at predicting assessment scores or students' GPA. Xing et al. [26] conducted both classification and regression predictions. For the regression task, they exclusively employed the Mean Absolute Percentage Error (MAPE) to assess the performance of two prediction models, namely LR and Support Vector Regression (SVR). The data samples were categorized by subcategories within six data features, aiming to identify the model that exhibited higher Mean Absolute Percentage Erro (MAPE) values across all categories, thereby indicating superior performance. Additionally, the authors segmented the MAPE results into minimum, maximum, and standard deviation based on multiple subcategory samples from each data feature. A higher standard deviation suggested a greater bias. To enhance model performance, they also implemented 10-fold cross-validation and GridSearch parameter tuning. Meanwhile, Hassan et al. [32] employed a comprehensive set of metrics, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Adjusted R-squared (Adj-R2), to evaluate their student GPA prediction model. Through a series of experiments comparing four different regression models, they identified the LassoCV Algorithm as the most effective method, particularly when combined with clustering and feature selection before regression tasks. This conclusion was supported by consistently low MAPE, RMSE, and Adj-R2 values compared to the other models.

5.6.2.3 Studies Use Statistical Testing

Khan et al. [76] employed a t-test based on F1-Score and conducted k-fold cross-validation (k=10) using the WEKA software, in order to compare the degree of effectiveness between classifier models. This validation process corroborated the earlier evaluation employing ML performance metrics (classified as group A in the Classification Task) and consistently demonstrated the highest F1-Score for their selected algorithm in the t-test.

Esteban et al. [63] employed the Wilcoxon signed-rank test to ascertain potential disparities in behaviors exhibited by algorithms utilizing transformed data representations. These representations were derived from two Multiple Instance Learning (MIL) based data representation techniques, namely SimpleMIL and WrappedMIL, in comparison to a conventional data representation. The study encompassed three distinct Wilcoxon signed-rank tests, each focusing on

different performance metrics including Accuracy, Recall (sensitivity), and Specificity within the framework of a 10-fold stratified cross-validation procedure. The findings revealed that MIL methods demonstrated notable enhancements over conventional approaches, with a confidence level p-value attaining up to 99% significance in terms of both Accuracy and Recall measures. In the context of specificity, crucial for identifying academically challenged students, while SimpleMIL yielded commendable results, MIWrapper encountered challenges in discerning the negative class relative to its counterparts. Through the integration of ML performance metrics (as delineated in Part A, Group D studies) and Wilcoxon signed-rank tests, the study substantiates the superior efficacy of MIL based methodologies in augmenting prediction models.

Cano and Leonard [37] present their MVGP classification methodology for detecting at-risk students and empirically establish its superiority over alternative algorithms. This validation is substantiated through the adoption of a comprehensive array of methodologies, encompassing six ML performance metrics (as delineated in 2.1.4), as well as two statistical tests employing these metrics. The Bonferroni–Dunn test, applied to discern statistically significant differences, reveals that whenever the rank of the methods diverges from the control algorithm by more than 1.3547, at a significance level of $\alpha = 0.05$, denoting 95% statistical confidence, MVGP exhibits significant disparities in all six ML performance metrics when compared to other algorithms. Furthermore, the Wilcoxon signed-ranked test, facilitating multiple pairwise comparisons between MVGP and alternative approaches, highlights noteworthy discrepancies in terms of Accuracy, Recall (sensitivity), G-Mean, and AUC. However, no significant distinctions were observed in specificity for DT J48 and MVMI, as well as in Kappa for MVMI.

5.6.2.4 Studies Use Empirical Testing

Mengash [77] exemplify the successful application of this method. Their investigation into the most influential features impacting student performance prompted the university dean to assign greater significance to these attributes in the admission process. After a year of implementation, there was a notable enhancement in the proportion of excellent and very good students compared to the previous year's cohort. In a contrasting endeavor, Hassan et al. [80] sought to employ data from different courses with a consistent data structure to assess their predictive model. However, contrary to their initial evaluation yielding an Accuracy of 92.73%, the testing results fell below 40% for all prediction models.

5.7 Synthesis Discussion

Our survey primarily identifies two prominent supervised Machine Learning tasks: classification tasks, which center on categorizing student performance into discrete levels or statuses; and regression tasks, focused on predicting assessment scores. It is noteworthy that one study conducted

experiments covering both Classification and Regression tasks, so we counted it as two separate instances.

The survey identifies four primary categories of evaluation methods used in the reviewed studies: (i) machine learning (ML) performance metrics for classification tasks, (ii) ML performance metrics for regression tasks, (iii) statistical testing, and (iv) empirical testing. All reviewed studies employ ML performance metrics as their primary evaluation approach, while a smaller subset additionally incorporates statistical testing or empirical validation.

Among classification-based studies, the most frequently reported performance metrics are Accuracy (83%), Recall (70%), Precision (57%), and F1-score (57%). To support comparative analysis, the studies are grouped according to the combinations of evaluation metrics they employ:

- **Group A (38%):** Studies relying exclusively on the four common classification metrics (Accuracy, Precision, Recall, and F1-score).
- **Group B (17%):** Studies combining the four common metrics with additional, less frequently used metrics, including ROC/AUC, Specificity, and Matthews Correlation Coefficient (MCC).
- **Group C (21%):** Studies employing a single common metric, such as Accuracy or Recall, either alone or in combination with non-metric-based evaluation approaches (e.g., Accuracy with standard deviation).
- **Group D1 (8%):** Studies integrating Accuracy and Recall alongside additional metrics, such as ROC/AUC or Specificity.
- **Group D2 (8%):** Studies using ROC/AUC as the sole evaluation metric.
- **Group RG (8%):** Studies evaluating regression-based tasks using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Adjusted R^2.

In addition to performance metrics, a minority of studies incorporate statistical testing methods, including the Wilcoxon signed-rank test, Bonferroni–Dunn test, and t-test (12.5%), or empirical testing procedures (8.3%) to further validate their findings.

5.8 Research Gap and Conclusion

Based on the systematic review of recent studies on at-risk student performance prediction in Learning Analytics, several recurring limitations were identified, indicating important opportunities for future research:

- **Lack of regression-based prediction tasks:** Existing studies predominantly focus on classification tasks for predicting student performance categories, while regression-based approaches remain underexplored. This limitation is particularly relevant in contexts where student performance is represented using continuous measures, such as numeric grades or GPA, and where grading schemes vary across courses.
- **Uncritical selection of evaluation methods:** Many studies do not provide sufficient justification for their choice of evaluation metrics or modeling approaches. This lack of methodological transparency can reduce confidence in the reported results and limit their applicability across institutional settings.
- **Limited use of statistical testing:** Despite the varying strengths and weaknesses of machine learning algorithms in handling data imbalance, overfitting, and noise, statistical significance testing is often omitted. The absence of such testing hinders robust model comparison and weakens conclusions regarding model superiority.
- **Neglect of bias–variance tradeoff considerations:** Few studies explicitly address the bias–variance tradeoff prior to model training and evaluation. Ignoring this issue can result in ambiguous model selection and reduced generalizability, which remains a persistent challenge in Learning Analytics research.
- **Scarcity of empirical validation:** Empirical testing using real-world institutional datasets is limited, largely due to the resource-intensive nature of Learning Analytics projects. The lack of empirical validation constrains the practical relevance and external validity of many proposed approaches.

Overall, these limitations highlight the need for more rigorous, transparent, and context-aware research methodologies. It is also notable that recent studies increasingly explore deep learning and neural network–based approaches, reflecting a growing trend toward advanced models that demonstrate strong predictive performance across diverse datasets.

Chapter 6

A Case Study: Applying CRISP-DM methodology with Learning Analytics and Machine Learning to Enhance Early Detection of At-risk Students in Higher Education

This chapter addresses Research Questions RQ3 and RQ4 through an empirical investigation of predictive modeling and feature interpretation in Learning Analytics. Institutional data from the SIS and LMS are integrated and used to develop and evaluate multiple machine learning classification models for at-risk student prediction. Model performance is assessed using established evaluation metrics, and the most effective approaches are identified across different stages of the academic term. In addition, feature importance analysis is conducted to enhance model interpretability and provide actionable insights into the key factors influencing student risk.

6.1 Introduction

The trend for remote learning technologies has skyrocketed in recent years, making it challenging for institutions to monitor students' learning progress effectively. The 2022 Canadian Student Wellbeing Study survey [6] reveals that 72% of higher education students have experienced a substantial decline in face-to-face interactions with their instructors. Notably, 40% have expressed an intention to withdraw from their academic institutions, and 32% reported feeling unsupported

by their respective educational institutions. This decline in traditional in-person communication is primarily attributable to the shift towards remote learning modalities in response to COVID-19, which has been proven to be accompanied by a significant issue of high dropout rates compared to traditional teaching [1]. On the other hand, the attrition of students from degree programs is a substantial concern for academic institutions due to its adverse effects on student well-being and the broader community. Additionally, it entails financial costs for educational institutions, such as the loss of cash inflows and significant societal costs [2]. In North America, LA has emerged as a powerful tool for leveraging institutional data to support teaching and learning, led by such institutions as the University of Michigan and the University of British Columbia [82]. Recognizing our university need to address the imperative of glsla, a comprehensive strategy is required to establish a robust data framework and implement LA tailored to the higher education context. Such the framework, coupled with ML focused on detecting at-risk students through key indicators influencing students' performance, offered a valuable tool for reducing student dropout rates and providing timely intervention to enhance student and institution success.

However, existing studies often lack methodological transparency, overlook important time-based performance indicators, and focus narrowly on final course failure as the sole risk criterion (see section 6.2 for more details). This study addresses these gaps by applying the CRISP-DM framework to develop a phase-based early warning system that integrates institutional (SIS) and course-level (LMS) data. We evaluate six classification algorithms and incorporate both historical and real-time learning features to predict at-risk students at multiple points during the course. The study aims to:

- Compare the effectiveness of machine learning models in different prediction phases
- Identify the most influential factors contributing to student risk.

Our goal is to offer a reproducible and interpretable approach for timely, data-informed interventions in higher education.

6.2 Gap Analysis and Solutions

Based on our systematic review in the previous chapter 5, we identified several recurring limitations: (1) a lack of justification for evaluation metric selection, which undermines the validity and interpretability of results; (2) limited attention to the bias-variance tradeoff, reducing model generalizability; (3) insufficient reporting of data preprocessing and feature engineering, compromising transparency and reproducibility; (4) the omission of statistical testing, weakening the reliability of model comparisons. To address these gaps, our study explicitly grounds its evaluation metric selection in the institutional context—specifically, the University of Ottawa's enrollment and grading system (Section 6.3.1)—and justifies its focus on Recall, Accuracy and Cohen's

Kappa as relevant measures of model effectiveness (Section 6.3.3.2). We employ `GridSearchCV` with 5-fold cross-validation to systematically manage model complexity and mitigate underfitting and overfitting, directly addressing the bias-variance tradeoff (Section 6.3.3.1). To promote reproducibility, we provide comprehensive documentation of our data preparation and feature engineering procedures (Section 6.3.2).

Moreover, we also address the limitation of defining at-risk students solely by final course grade —an approach often impractical due to missing grades from early withdrawals or disengagement. Instead, we propose a mapping rule that captures a wider range of at-risk scenarios, including absence, early dropouts, and late withdrawals,etc (table 6.1).

Finally, while only 39% of reviewed studies use time-based features [37, 60, 62, 64, 65, 68, 70, 75, 76], our findings highlight their value in early detection(Sections 6.4.2). By incorporating cumulative and phase-specific performance metrics, our model predicts risk at multiple stages—course start, 30%, 50%, and 75%—enabling timely intervention efficiently(see results at section 6.4.1)

6.3 Research Methodology

This paper adopts the CRISP-DM- a framework to ensure a structured and transparent research process with extensibility in mind to support future enhancements. CRISP-DM builds upon and elaborates the original KDD framework, which Fayyad et al. [12] defined as the overall process of transforming large volumes of raw data into valid, novel, useful, and understandable knowledge. While data mining plays a central role, KDD also emphasizes important steps such as data preparation, selection, cleaning, incorporation of prior knowledge, and proper interpretation of results to derive meaningful insights. CRISP-DM further extends this approach by organizing the process into six interconnected phases: Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment [13]. During the past two decades, this methodology has become widely adopted and is often regarded as a "de facto standard for data mining projects," particularly in the health and education sectors [14]

The following sections will detail each phase of the CRISP-DM methodology, with the exception of the Deployment phase, which is beyond the scope of this paper and will be addressed in future work. Figure 6.1 depicts the six steps of the CRISP-DM framework and illustrates their adaptation to the context of early prediction of at-risk students in higher education. The phases in CRISP-DM are not strictly linear; moving back and forth between them is often necessary, depending on the outcomes of each step [15]. The colored arrows in Figure 6.1 represent common dependencies, while the grey arrow and circle reflect the cyclical nature of data mining—where lessons learned from one project inform and improve future iterations.

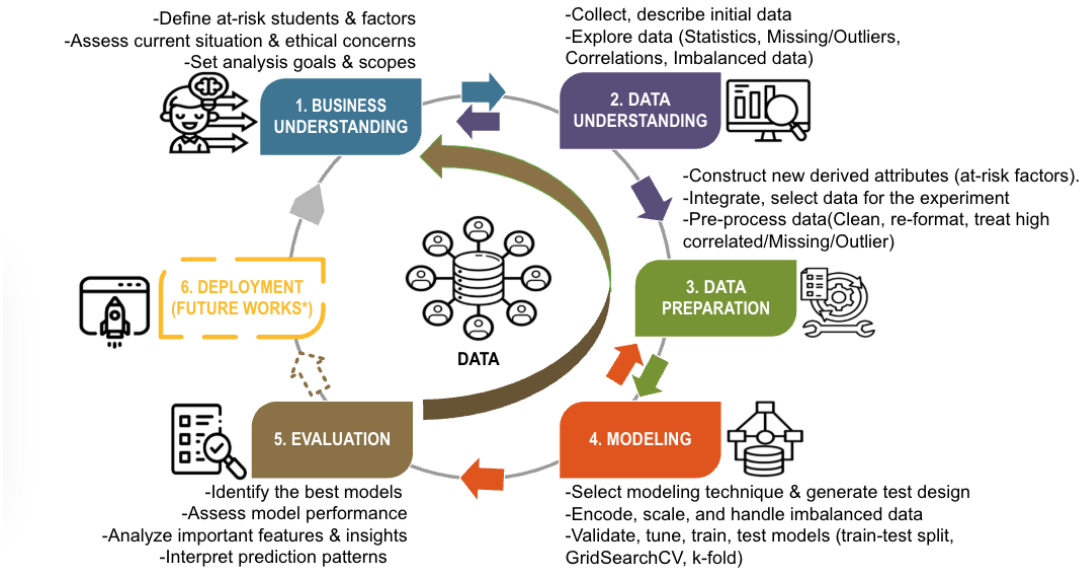


Figure 6.1: CRISP-DM Framework For At-risk Student Prediction Project (Deployment phase excluded from current study and will be addressed in future work). Adapted from CRISP-DM guidelines [15]

6.3.1 Business Understanding

This initial phase involves identifying the project's goals and requirements from a business standpoint and translating them into a defined data mining problem [15].To translate our objective (section 6.1) into a data mining task, the study focuses on the following: (1) defining at-risk students; (2) assessing relevant resources and ethical considerations; and (3) determining the analysis goal and scope.

6.3.1.1 Define at-risk students

To define at-risk students, we must understand how the enrollment process and grading system works.

At the start of each academic term, students register for various courses, but not all students complete them successfully. At the University of Ottawa, students are allowed to enroll in or withdraw from courses according to specific deadlines [83]. Students can enroll in courses up to approximately 15% into the course duration. They can withdraw and receive a full refund if they do so by around the 25% mark of the course's duration. Withdrawals are still allowed up to approximately 80% of the course duration, but without a refund and without affecting the

student's GPA. After this point, students are no longer allowed to withdraw, and if they stop participating or fail to meet course requirements, a failing grade is recorded that impacts their GPA. Regarding the grading system [84], the University of Ottawa's official grading system is alphanumeric. To pass most undergraduate courses, students need at least a grade of D.

In this study, we group course outcomes into two categories: Passed and At-Risk. A student is considered "Passed" if they earn a grade of C or higher. We also account for students who have no final grade but do have enrollment records. If they withdrew before 25% of the course was completed, we label them as "Dropped Before Attempt"—these students are not considered at risk, as their withdrawal may have been a thoughtful decision made early. On the other hand, students who withdrew later in the course are labeled "Dropped After Attempt." Although these students avoided a failing grade, they had already invested time and effort and likely faced challenges. Therefore, we consider them part of the At-Risk group as well. See table 6.1 for more details.

Table 6.1: Mapping Rules for Course Results and Model Target Labels

Condition / Rule	**Mapping1: Grade Category (For future research)**	**Mapping2: Model Target Label**
Condition 1: Course Grade Exists		
Grade = A+, A	Excellent	Passed
Grade = A-, B+, B, C+, C, D,D+,P, S	Passed	Passed
Grade = F, E, EIN, NS	Failed	At-risk
Grade = DR, ABS, INC	Dropped/Absent	At-risk
Grade = CR, Q, AUD,NC	Audit/No Units/OtherInstitution	*Filtered out*
Grade = DNW, CTN, DFR, AEC	Continue/Deferred	*Filtered out*
Grade = NNR	Not Updated	*Filtered out*
Condition 2: Course Grade is Missing or Empty		
Enrollment = D and Attempt = Y or I	Drop After Attempt	At-risk
Enrollment = W (Waiting)	No Attempt	*Filtered out*
Enrollment = E and Attempt = N	No Attempt	*Filtered out*
Enrollment = D and Attempt = N	Drop Before Attempt	*Filtered out*
Enrollment = E and Attempt = I	Not Updated	*Filtered out*

6.3.1.2 Access possible at-risk factors and ethical concerns

Our preliminary work [82] found that the available student data landscape is extensive and diverse. However, collecting all relevant data at the initial stage was not feasible due to both technical complexity and ethical permission constraints. For this study, we focused on gathering essential information, including student demographics and course enrollment records. The at-risk students

factors were scoped around academic performance—such as course and assignment grades—and enrollment activity, using data obtained from the University of Ottawa's Student Information System (SIS) and the Brightspace Learning Management System (LMS). These data sources are described in greater detail in Section 6.3.2 (Data Understanding).

All data used in this study are non-identifiable which were submitted for review and formally approved by the University of Ottawa's Research Ethics Board (REB) under ethics file number H-04-24-10374.

6.3.1.3 Determine the data mining goals and scopes

This study aims to build a **binary classification model** to predict whether a student will be **At-Risk (label 1)** or **Passed (label 0)** in a course. Predictions are generated at multiple stages of the course to support timely interventions:

- **Early stage (15% course duration)**: Uses historical student and course data from the *Student Information System (SIS)*, including demographics and academic history.
- **Mid and late stages (30–75% course duration)**: Incorporates graded items (assignments, quizzes, etc) from the *Brightspace Learning Management System (LMS)*, allowing for more refined predictions based on student performance.

Each record corresponds to a student's participation in a course and is labeled according to the academic and withdrawal criteria outlined in Section 3.1.1. The model is primarily evaluated using accuracy and recall, with particular emphasis on minimizing false negatives—cases where the model incorrectly predicts a student will pass (label 0) when they are actually at risk (label 1). This ensures that as many at-risk students as possible are identified and not overlooked.

6.3.2 Data Understanding & Data Preparation

The Data Understanding and Preparation phases involve exploring, transforming, and organizing raw data into a format suitable for modeling [15].Figure 6.2 illustrates the overview of the data preparation process in data layers adapted from our Learning Analytics System Architecture proposed in [82]

6.3.2.1 Initial Dataset Overview

Table 6.2 summarizes the 17 relational tables used in this study , sourced from two institutional systems: The first source- SIS dataset- encompasses academic records for 11,200 undergraduate

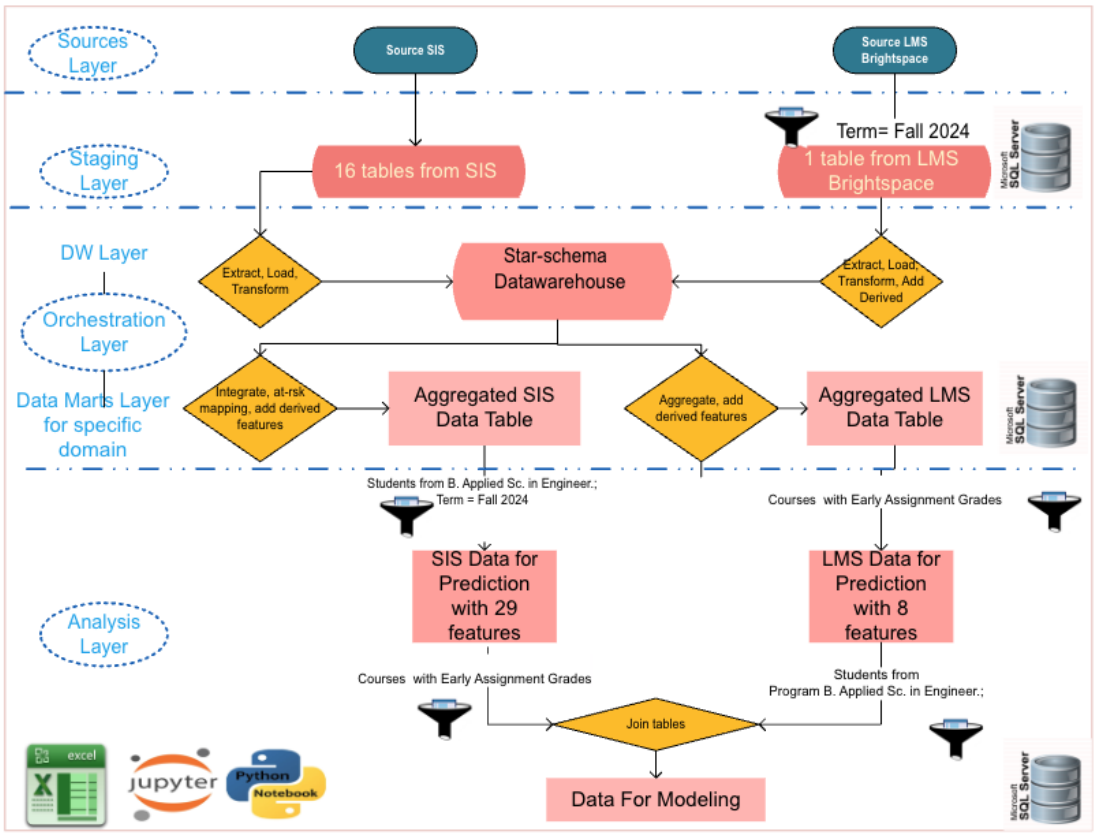


Figure 6.2: Data Preparation Process. Data layers are adapted from Learning Analytics System Architectures proposed in our related work [82]

students, including their enrollment in 1,405 unique courses spanning from Fall 2011 to Fall 2024. These records capture both final class grades and course withdrawal outcomes. The second source- the Brightspace LMS dataset- provides detailed assessment-level data, comprising all available assignment submissions recorded only in Fall 2024 across the entire University of Ottawa

6.3.2.2 Extract, Transform, Load (ETL)Process

The original dataset, composed of normalized relational tables, was extracted and restructured into a star-schema model to support efficient querying and simplified reporting. This dimensional schema includes one central fact table surrounded by multiple dimension tables, where the fact table stores measurable events (e.g., course enrollments and results), and dimension tables provide contextual information (see Figure 6.3 for schema details). This transformation from a normalized

Table 6.2: Summary of Initial Data Tables from SIS and LMS Sources

#	Table Name	Description	Data Scope / Filters	Source
1	Academic_Career_d	Defines student career types (e.g., undergrad, grad).	Type = Undergraduate	SIS
2	Academic_Plan_d	Lists majors, minors, or specializations.	23 undergraduate majors in Faculty Of Engineering	SIS
3	Academic_Program_d	Describes academic programs offered.	Program='B.Applied Sc - Engineering'	SIS
4	Class_Term_d	Details classes offered by term.	19,496 undergraduate classes	SIS
5	Country_d	Lookup for country names and codes.	269 country records	SIS
6	Course_d	Catalog of courses with titles and credits.	1,405 undergraduate courses	SIS
7	Date_d	Calendar dimension for time-based queries.	01/09/2011 to 31/12/2024	SIS
8	Degree_d	Lists degree types (e.g., B.A., M.Sc.).	23 degree types	SIS
9	Faculty_Service_d	Teaching/service assignments for faculty.	Faculty service= Faculty of Engineering	SIS
10	Institution_d	Institutional or campus metadata.	Institution= University of Ottawa	SIS
11	Modality_d	Course delivery modes (e.g., online, hybrid).	9 delivery modes at uOttawa	SIS
12	Student_d	Core student demographic information.	11,200 students registered for "B.Applied Sc-Eng." in first term	SIS
13	Student_Term_f	Student term-level enrollment summary.	All term enrollment activities of students in table (12)	SIS
14	Subject_d	Academic subject areas (e.g., MATH, PHYS).	20 undergraduate subjects	SIS
15	Student_Program_Enrollment_f	Tracks student enrollment by academic program.	1,461,397 activity records	SIS
16	Student_Class_Term_Enrollment_f	Tracks student enrollment in classes by term.	6,715,461 activity records	SIS
17	Brightspace_AllGrades	Graded items (Assignments, Quizzes, etc) records	1,609,547 grade records in Term 2024 Fall	LMS

to a denormalized structure improves performance and supports the analytical tasks required for this study.

Given the objectives of this study, particular attention was paid to the transformation of the CourseGradeOfficial field, which serves as the target label in the classification model developed in later phases. The transformation process followed a set of mapping rules (see Table 6.1). All transformation logic was implemented in Microsoft SQL and executed within a Visual Studio Code environment. The SQL scripts developed for this stage include the creation of dimension tables, fact tables, and aggregation tables corresponding to each data warehouse layer illustrated in Figure 6.3. These scripts are provided in Appendix B.

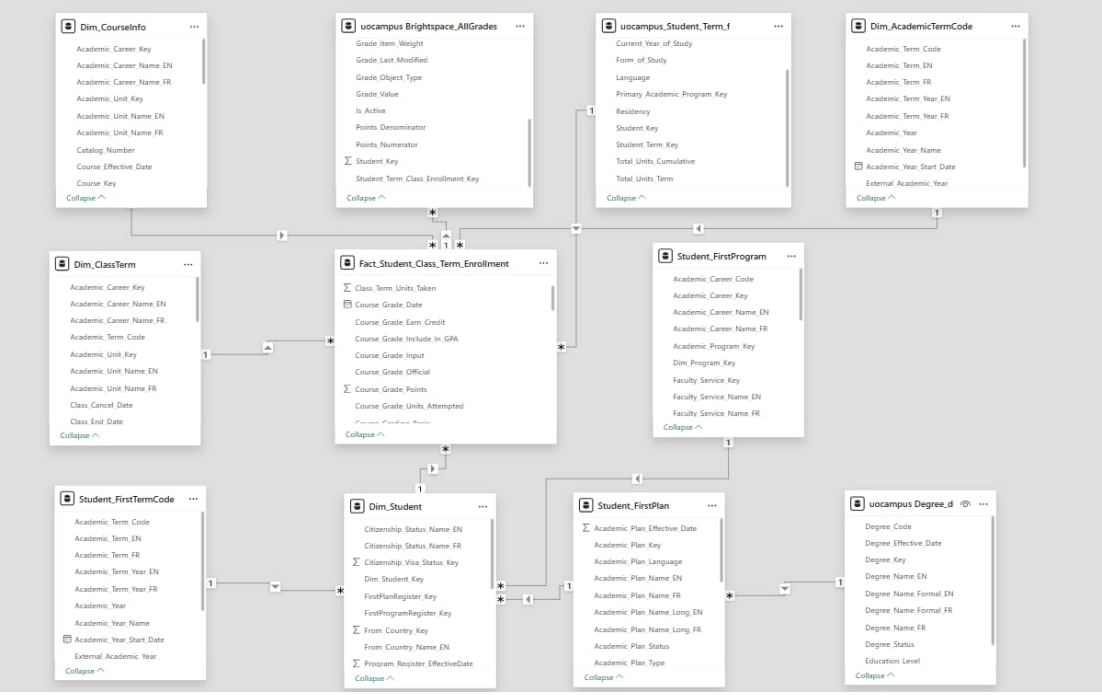


Figure 6.3: Star schema representation of the academic data model used in this study, centered around student class-term enrollment as the fact table, with associated dimension tables from SIS and LMS systems. Source: authors.

6.3.2.3 Feature Engineering and Selection

Feature engineering enhances data representation by constructing informative variables from raw data, while feature selection reduces dimensionality by retaining the most relevant features to improve generalization and interpretability [85, 86]. In LA contexts, where educational datasets are often high-dimensional and noisy, these processes are essential for developing robust and reliable predictive models [87].

To support the binary classification of at-risk students, all data features included in the requested list (as mentioned in Chapter 4, Section 4.4.1) and provided by the uOttawa Data Hub team were initially selected. These features comprise both raw attributes from the SIS and engineered metrics derived from students' academic records and LMS activity data. Given the limited availability of institutional data, an inclusive feature selection strategy was adopted at this stage to retain as much potentially relevant information as possible. Features that did not demonstrate significant predictive contribution were subsequently removed based on findings from the exploratory data analysis, as discussed in the following section. In addition, time-aware features were constructed from historical academic records, while assignment-level performance

data from Brightspace LMS was used to generate learning progress features at three course milestones: Phase 1 (30%), Phase 2 (50%), and Phase 3 (75%). Table 6.3 summarizes all features used in model development. Feature engineering was implemented using Microsoft SQL within Visual Studio Code. The corresponding scripts are provided in the *Analysis Layer - Retrieval Queries for Machine Learning Projects* section (Section **??**).

To ensure data quality and alignment across sources, the dataset was filtered based on the following criteria:

- **Program:** Only undergraduate students enrolled in the *Bachelor of Applied Science – Engineering* program were included, representing approximately 73% of engineering undergraduates faculty in Fall 2024.
- **Term:** Courses from *Fall 2024* were selected to match the term covered by Brightspace LMS records. Historical SIS data, however, was retained and transformed into time-based features (see Table 6.3).
- **LMS Course Activity:** Only Brightspace-enabled courses with assignment grade data available before 75% of course completion were included. This ensured the relevance of LMS engagement features and consistency with SIS course records.

6.3.2.4 Exploratory Data Analysis

The final dataset used for model development consists of **6,820 student-registeredCourse records** and **37 features** (refer to Table 6.3). This section presents a comprehensive exploratory data analysis (EDA) to better understand the structure, relationships, and quality of the dataset. The exploratory data analysis (EDA) process was implemented using the Python programming language and is documented in Appendix C: Phase1: Data Exploration (Section C.1). All outputs generated during this stage are presented in a Jupyter Notebook, which has been exported to PDF format and is available at: https://github.com/mytu007/uOttawa_ThesisProject/blob/main/Expoloration_AllPhases.ipynb.

The EDA process includes the following key components:

Descriptive Statistics: Summary statistics were computed to understand the central tendency and dispersion of numeric features, as well as the distribution of categorical variables. Figure 6.4a presents the count, minimum, maximum, mean, standard deviation, and quartiles for all numeric features, while Figure 6.4b summarizes the categorical features by showing the number of unique values, the most frequent category (mode), and its frequency.

As shown in Figure 6.4a, some derived features (e.g., *Cumm PreviousTermEarnedUnitRates*, *Cumm PreviousTermPercentComplete*) contain missing values, primarily because

Table 6.3: Summary of Features Used for Modeling

#	Feature Name	Description	Data Type
Raw Features Integrated or Transformed from SIS Dimensional & Fact Tables			
1	Student Gender	Student Gender	Binary
2	Modality Description	The mode of instruction for the course	Nominal
3	Course Code	The unique identifier for the course	Nominal
4	Subject Name EN	The English name of the subject associated with the course	Nominal
5	Class Language	The primary language in which the class is taught	Nominal
6	Diploma Description EN	The name of the diploma or degree the student is pursuing	Nominal
7	Study Field	Student Study field of registered program	Nominal
8	Form of Study	Form of study of current record's term	Nominal
9	Academic Load	Academic Load Type of current record's term	Nominal
10	Total Units Term	No Of enrolled units in current record's term	Numeric
11	Current Year of Study	Student current study year of current record	Ordinal
12	Current Residency	Student current residency at the moment of current record	Nominal
13	Class Season	Course Season	Nominal
14	Target Label	Model Label prediction target in 3.2.1	Binary
Time-based Features Derived from SIS Fact Table			
15	Start Residency Status	Student residency status in the first term	Nominal
16	Student Term Number	Student current term number of study	Ordinal
17	From Country Name EN	Student residential country in the first term	Nominal
18	Current Age	Student age at the moment of current record	Numeric
19	Course No Of Attempt	The number of times a student has attempted the course	Numeric
20	Cumm Previous Term Earned Unit Rates	The cumulative percentage of units successfully earned	Numeric
21	Cumm Previous Term Percent Complete	The cumulative percentage of course completed	Numeric
22	Cumm Previous Term GPA	The student's cumulative Grade Point Average	Numeric
23	Cumm Previous Term Passed Rates	The cumulative percentage of courses passed	Numeric
24	Cumm Previous Term At Risk Rates	The cumulative percentage of at-risk courses	Numeric
25	Cumm Previous Term Dropped Rates	The cumulative percentage of dropped courses after attempt	Numeric
26	Course Previous Term AVG Dropped Rates	The average drop rate of the current course in prior terms	Numeric
27	Class Term Units Taken	The total number of units the student is taking	Numeric
28	AVG Previous Term No Of Registered Units	The average number of units the student registered in one term	Numeric
29	Student Start Season	The academic season in the first term	Nominal
Time-based Features Derived from LMS Fact Table (Brightspace)			
30	Grading System	Indicates if the grading was weight-based or point-based	Nominal
31	AVG Grade Phase 1	The average grade of graded items completed up to Phase 1	Numeric
32	AVG AVG Grade Group Phase 1	The average of group items averages up to Phase 1	Numeric
33	AVG Grade Phase 2	The average grade of graded items completed up to Phase 2	Numeric
34	AVG AVG Grade Group Phase 2	The average of group items averages up to Phase 2	Numeric
35	AVG Grade Phase 3	The average grade of graded items completed up to Phase 3	Numeric
36	AVG AVG Grade Group Phase 3	The average of group items averages up to Phase 3	Numeric
37	Class NoOfStudent	Number of distinct students (excluding any filters in SIS) have graded records in the course	Numeric

they correspond to students in their first academic term with no prior cumulative data. These features remain important for analyzing distributions and treating outliers in the pre-processing phase. Figure 6.4b highlights a class imbalance, with approximately 80% of students labeled as *Passed*, indicating a skewed distribution in the target variable.

	count	mean	std	min	25%	50%	75%	max
Student_Term_Number	6820.0	4.37	4.51	1.00	1.00	1.00	7.00	33.00
Current_Age	6820.0	19.73	2.62	15.00	18.00	19.00	21.00	51.00
Total_Units_Term	6820.0	14.31	2.38	0.00	12.00	15.00	15.00	21.00
Course_NoOfAttempt	6820.0	1.05	0.26	1.00	1.00	1.00	1.00	5.00
Cumm_PreviousTermEarnedUnitRates	3206.0	100.67	11.33	8.00	100.00	100.00	105.00	150.00
Cumm_PreviousTermPercentComplete	3237.0	46.60	24.96	0.00	23.00	48.00	66.00	233.00
Cumm_PreviousTermGPA	3206.0	6.81	1.69	0.23	5.58	6.70	8.11	10.00
Cumm_PreviousTermPassedRates	3237.0	88.76	14.11	0.00	82.00	93.00	100.00	100.00
Cumm_PreviousTermAtRiskRates	3237.0	11.22	14.11	0.00	0.00	6.00	18.00	100.00
Cumm_PreviousTermDroppedRates	3237.0	0.05	0.82	0.00	0.00	0.00	0.00	24.00
Course_PreviousTerm_AVGDroppedRates	6799.0	9.86	7.10	0.00	4.00	8.00	16.00	60.00
Class_Term_Units_Taken	6820.0	2.67	0.94	0.00	3.00	3.00	3.00	3.00
AVG_PreviousTermNoOfRegisteredUnits	3237.0	10.93	2.82	0.00	9.00	10.88	13.00	18.00
Avg_Grade_Phase3	6820.0	78.82	18.06	0.00	71.00	83.00	92.00	112.00
Avg_Avg_GradeGroup_Phase3	6820.0	78.82	18.06	0.00	71.00	83.00	92.00	112.00
Class_NoOfStudent	6820.0	645.13	1513.33	2.00	85.00	134.00	177.00	5626.00
Avg_Avg_GradeGroup_Phase1_Adjusted	6820.0	0.68	0.40	0.00	0.41	0.89	1.00	1.10
Avg_Avg_GradeGroup_Phase2_Adjusted	6820.0	0.75	0.31	0.00	0.69	0.87	0.96	1.07
Avg_Grade_Phase1_Adjusted	6820.0	0.68	0.40	0.00	0.41	0.89	1.00	1.10
Avg_Grade_Phase2_Adjusted	6820.0	0.75	0.31	0.00	0.69	0.87	0.96	1.07

(a) Numeric Features

	count	unique	top	freq
Student_Gender	5840	4	M	4226
From_Country_Name_EN	5840	7	Canada	4779
Modality_Description	5840	3	In Person	5531
Subject_Name_EN	5840	15	Engineering	1404
Class_Language	5840	3	EN	4503
Study_Field	5840	7	MCGE	1438
Form_of_Study	5840	2	ENRL	5819
Academic_Load	5840	2	F	5575
Current_Year_of_Study	5840	5	01	2855
Current_Residency	5840	5	CDN	4431
Target	5840	2	Passed	4719
Student_Start_Season	5840	2	Fall	5778
Class_Season	5840	1	Fall	5840
GradingSystem	5840	2	Weight-based	4465

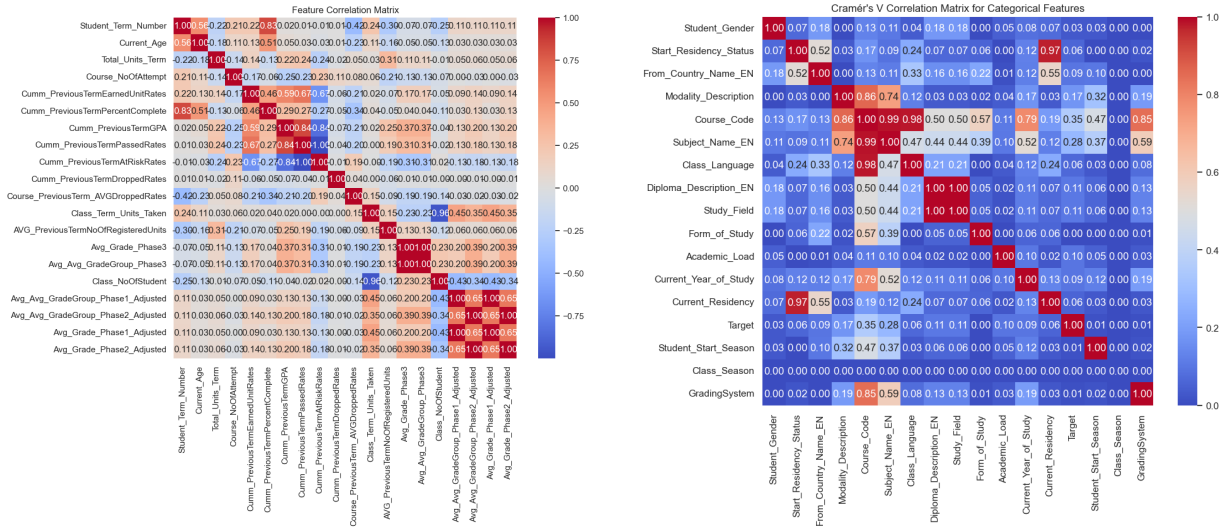
(b) Categorical Features

Figure 6.4: Descriptive Statistics

Correlation Analysis: *Pearson's correlation coefficient* [17] was used to evaluate linear relationships among numeric features, as shown in Figure 6.5a. Several features demonstrated strong multicollinearity ($r \geq 0.83$), particularly *Cumm PreviousTermPercentComplete*, *Cumm PreviousTermGPA*, and *Cumm PreviousTermPassedRates*, which were highly correlated with *Student Term Number* and *Cumm PreviousTermAtRiskRates*. These redundant features will be removed in the preprocessing phase. For categorical features, *Cramér's V* [18] - which is a statistical measure derived from the Pearson's Chi-Square statistic [88]- was used to assess associations between categorical feature pairs (see Figure 6.5b). Strong associations were observed between *Course Code*, *Start Residency Status*, and *Diploma Description EN* with other categorical features such as *Modality Description*, *Current Residency*, and *Study Field*, respectively. To reduce redundancy, we will drop *Course Code*, *Start Residency Status*, and *Diploma Description EN* in the next preprocessing step.

Outlier Analysis: For numeric features, outliers were identified using the Interquartile Range (IQR) method [19]. Q1 (25th percentile) and Q3 (75th percentile)- as shown in Figure 6.4a - were used to compute the IQR as $\text{IQR} = Q3 - Q1$. Outlier thresholds were then calculated using a standard multiplier of 1.5: values below $Q1 - 1.5 \times \text{IQR}$ or above $Q3 + 1.5 \times \text{IQR}$ were flagged as outliers. In the next preprocessing phase, these values will be capped at the respective lower or upper bounds.

For categorical features, rare values were identified through frequency analysis. A category is considered rare if its relative frequency is below the given threshold (now default is at 1%). Features such as *From_Country_Name_EN* (where most students are from Canada) and *Subject_Name_EN* (with Engineering being the most common) exhibited a long tail of infrequent values. Additional rare entries were found in *Student_Gender* ('U'), *Form_of_Study* ('CO', 'EIP'), *Current_Year_of_Study* ('00', 'D'), *Current_Residency* ('REFUG', 'ACH-D', 'OTHER'), and *Student_Start_Season* ('Winter', 'Spring-Summer'). These rare values will be all replaced by "Others" during the next preprocessing phase to improve model robustness.



(a) Pearson's Correlation Matrix For Numeric Features

(b) Cramér's V Correlation Matrix for Categorical Features

Figure 6.5: Correlation Analysis

6.3.3 Modeling

This section outlines the machine learning algorithms employed to develop the predictive models, along with the data preprocessing strategies, techniques for handling class imbalance across different algorithms, and the procedures used for model validation and testing. Finally, the predictive models are evaluated using performance metrics that are most relevant to the data mining task defined in Section 3.1.3.

The modeling process for each dataset, including Dataset 1 (without Brightspace data) and Phases 1, 2, and 3, was implemented using the Python programming language and is docu-

mented in Appendix C under *Phase 2: Prediction Model* (Section C.2). All outputs generated during this stage are presented in a Jupyter Notebook, which has been exported to PDF format and is available at: https://github.com/mytu007/uOttawa_ThesisProject/blob/main/model_data2249_Phase3.ipynb.

6.3.3.1 Model Setup and Processes

Numerous classification algorithms have been proposed in the literature, ranging from traditional models (such as Decision Trees and Logistic Regression, etc)to more advanced techniques (such as Multiview Genetic Programming, Deep Learning, Artificial Neural Networks, and Convolutional Neural Networks, etc). While recent research highlights the potential of these advanced models, we observed that traditional algorithms continue to deliver strong predictive performance in similar educational datasets—often exceeding 70% accuracy [60, 62, 64, 67, 70, 73, 74, 76, 78, 81]. Moreover, classical models offer well-established interpretability techniques, which are particularly valuable in educational contexts where transparency and explainability are essential. For these reasons, we adopted traditional machine learning algorithms as strong baselines for this study, with the intention to explore more complex models in future work. Six classification algorithms were selected for this study, including Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Support Vector Machine (SVM), and Gaussian Naïve Bayes (Gaussian), all implemented using the scikit-learn library [21]. Additionally, Gradient Boosting was implemented using the XGBoost library [89].

To optimize model performance, hyperparameter tuning [20] was performed using the GridSearchCV function from Scikit-learn's model_selection module [21]. This approach exhaustively searches through specified parameter combinations and evaluates each using 5-fold cross-validation [90]. In each fold, the model is trained on four subsets and validated on the remaining one, cycling through all five folds. The evaluation metric used for selection was the recall score, given its importance in minimizing false negatives in at-risk student prediction.

To address class imbalance during model training, appropriate strategies were applied based on the algorithm. Tree-based models such as Decision Tree and Random Forest used class_weight="balanced" in scikit-learn, while XGBoost leveraged the scale_pos_weight parameter, calculated from the ratio of negative to positive samples in the training data. For models without built-in imbalance handling—such as Logistic Regression, SVM, and Gaussian Naïve Bayes—SMOTE (Synthetic Minority Over-sampling Technique [91]) from the imblearn library was used to oversample the minority class. Additionally, StandardScaler() from scikit-learn was applied to normalize numeric features for these models, which rely on distance or gradient-based optimization, unlike tree-based models that split data based on feature thresholds (e.g., Is feature $X > 4.5$?) and are unaffected by feature scaling.

Figure 6.6 illustrates the end-to-end workflow for model development. The process begins with data preprocessing, which includes removing highly correlated features (Section 3.2.4), treating

outliers using the interquartile range (IQR), replacing rare categorical values with "Other" (Section 3.2.4), and imputing missing values using the mean to minimize performance bias. The processed dataset is then divided into four subsets based on prediction phases (Phase 0 to Phase 3). Phase 0 contains only features from the SIS dataset. Phases 1 through 3 include all SIS features along with corresponding LMS features available at each phase (e.g., Avg_Grade_Phase1, Avg_Avg_GradeGroup_Phase1 for Phase 1), progressively reflecting students' learning progress. Each dataset is used to train six classification models using GridSearchCV with 5-fold cross-validation, optimizing for the recall score. Categorical features are encoded using OneHotEncoder from scikit-learn prior to an 80/20 train-test split.

To address class imbalance, scale_pos_weight is applied for XGBoost, class_weight='balanced' (setting to tells the model to give more weight to the minority class during training) is used for Random Forest and Decision Tree, and SMOTE is applied for other non–tree-based models, along with StandardScaler() for numeric feature normalization.

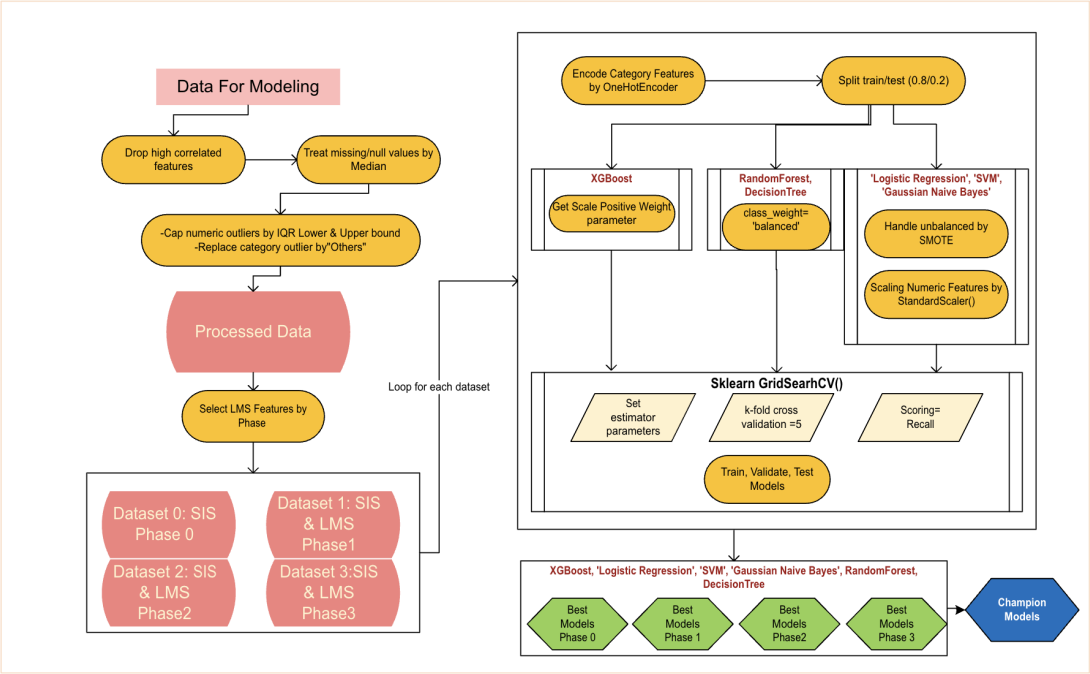


Figure 6.6: Pre-processing and Model Training Processes using Python Jupyter Notebook

Finally, GridSearchCV returns the best estimator for each model and phase, which is then used in subsequent analysis. Table 6.4 represents all algorithms, the library and its best estimators.

Table 6.4: Best Hyperparameters identified by GridSearchCV in This Study

Python Library	Model	Best Parameters
scikit-learn	Random Forest	{class_weight: 'balanced', max_depth: 10, max_features: 'sqrt', min_samples_leaf: 8, min_samples_split: 2, n_estimators: 100}
scikit-learn	Decision Tree (CART)	{class_weight: 'balanced', max_depth: 10, min_samples_split: 5}
XGBoost	XGBoost	Phase 2&3: {learning_rate: 0.01, max_depth: 3, n_estimators: 100, scale_pos_weight: 4.208} Phase 0&1: {learning_rate: 0.1, max_depth: 3, n_estimators: 100, scale_pos_weight: 4.208}
scikit-learn	Logistic Regression	{C: 0.1, penalty: 'l2'}
scikit-learn	SVM	{C: 0.1, kernel: 'poly'}
scikit-learn	Gaussian Naive Bayes	{var_smoothing: np.float64(1.0)}

6.3.3.2 Evaluation Metrics

In this study, Recall is prioritized as the key evaluation metric, as it reflects the model's ability to correctly identify students who are genuinely at risk of failure (true positives). Failing to detect these students (false negatives) could result in missed opportunities for timely intervention. However, relying solely on recall may increase false positives, where students not truly at risk are incorrectly flagged—potentially leading to unnecessary concern, resource misallocation, and reduced confidence in the system. To provide a more balanced view, we also report Accuracy as a general indicator of overall prediction correctness, and Precision, which indicates the proportion of flagged students who are actually at risk. While there is often a trade-off between precision and recall, analyzing both helps us better understand the practical implications of our model's performance. Lastly, Cohen's Kappa [92] is a helpful statistical measure for the interpretation of classification models so that we can learn about the performance gain over random guessing [93].

All metrics were computed using functions from the sklearn.metrics module [21].

- **Accuracy** measures the proportion of total correct predictions:

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

  where TP, TN, FP, and FN represent the four fundamental components of a confusion matrix (see Table 6.5) used to evaluate classification models:

  - True Positive (TP): The number of at-risk students correctly identified as at-risk by the model.
  - True Negative (TN): The number of students correctly identified as not at risk.

– False Positive (FP): The number of students incorrectly identified as at risk (predicted at-risk, but actually not at risk).
– False Negative (FN): The number of at-risk students incorrectly identified as not at risk (missed by the model).

Table 6.5: Confusion Matrix

		Actual	
		Positive (1)	**Negative (0)**
Predicted	**Positive (1)**	True Positive (TP)	False Positive (FP)
	Negative (0)	False Negative (FN)	True Negative (TN)

- **Recall** (Sensitivity or True Positive Rate) measures the proportion of actual positive cases that are correctly identified. In our context, it indicates how many at-risk students are successfully detected out of all students who are truly at risk. Conversely, it reflects the extent to which at-risk students may be missed by the model.

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

- **Precision** measures the proportion of predicted positive cases that are actually correct. In our context, it reflects how many of the students flagged as at risk are truly at risk, and helps indicate the extent to which non–at-risk students may have been incorrectly classified.

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

- **Cohen's Kappa** evaluates agreement between predicted and actual classifications. A kappa of 1 indicates perfect agreement, whereas a kappa of 0 indicates agreement equivalent to chance:

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

where p_o is the observed agreement and p_e is the expected agreement by chance; Table 6.6 describes interpretation of Kappa scores.

6.4 Results and Discussion

This section presents the experimental results used to identify the best-performing models for predicting students at risk of course failure or dropout. Following this, we examine feature

Table 6.6: Interpretation of Cohen's Kappa Values [94]

Kappa Value	Interpretation
$\kappa < 0$	Poor agreement
$0.00 \leq \kappa \leq 0.20$	Slight agreement
$0.21 \leq \kappa \leq 0.40$	Fair agreement
$0.41 \leq \kappa \leq 0.60$	Moderate agreement
$0.61 \leq \kappa \leq 0.80$	Substantial agreement
$0.81 \leq \kappa \leq 1.00$	Almost perfect agreement

importance to better understand model behavior and uncover which student-related factors most strongly influence performance and contribute to at-risk predictions. Figure 6.7 illustrates the performance comparison of six classification algorithms. In addition, table 6.6 illustrates the interpretation of Cohen's Kappa values by Landis and Koch [94] and will be used to compare the trustworthiness between models in the next analysis section.

6.4.1 Best Performing Model

As discussed in Section 3.1.3, Recall is prioritized in this study for selecting the best-performing models, as it reflects the model's ability to correctly identify at-risk students. Based on figure 6.7 , three algorithms in all phases—Gaussian Naïve Bayes, Random Forest (RF), and XGBoost—consistently achieved recall values exceeding the average threshold of 72.7% (yellow line), which we consider a strong performance benchmark. However, Gaussian Naïve Bayes was excluded from further analysis due to its persistently low Accuracy and Cohen's Kappa values across all phases. Consequently, only Random Forest and XGBoost are considered leading candidates for the best-performing models.

Two similar trends were observed for the champion candidates. First, both RF and XGBoost achieved their highest recall in Phase 0 (using only SIS data), followed by a decline in Phases 1 and 2, then a recovery in Phase 3 as more LMS data was integrated. This suggests that partial LMS data (Phases 1 and 2) may introduce noise or inconsistencies, whereas full LMS data in Phase 3 contributes to model stabilization and improvement. Second, both Accuracy and Cohen's Kappa increased steadily from Phase 0 to Phase 3, indicating improved overall reliability and reduced false alerts as more comprehensive LMS data was introduced. Table 6.7 describes details analysis for each phase.

6.4.2 Predictor Importance

Figure 6.8 presents the feature importance analysis across the four course phases, highlighting how the model's reliance on different features evolves as more data becomes available. In the early

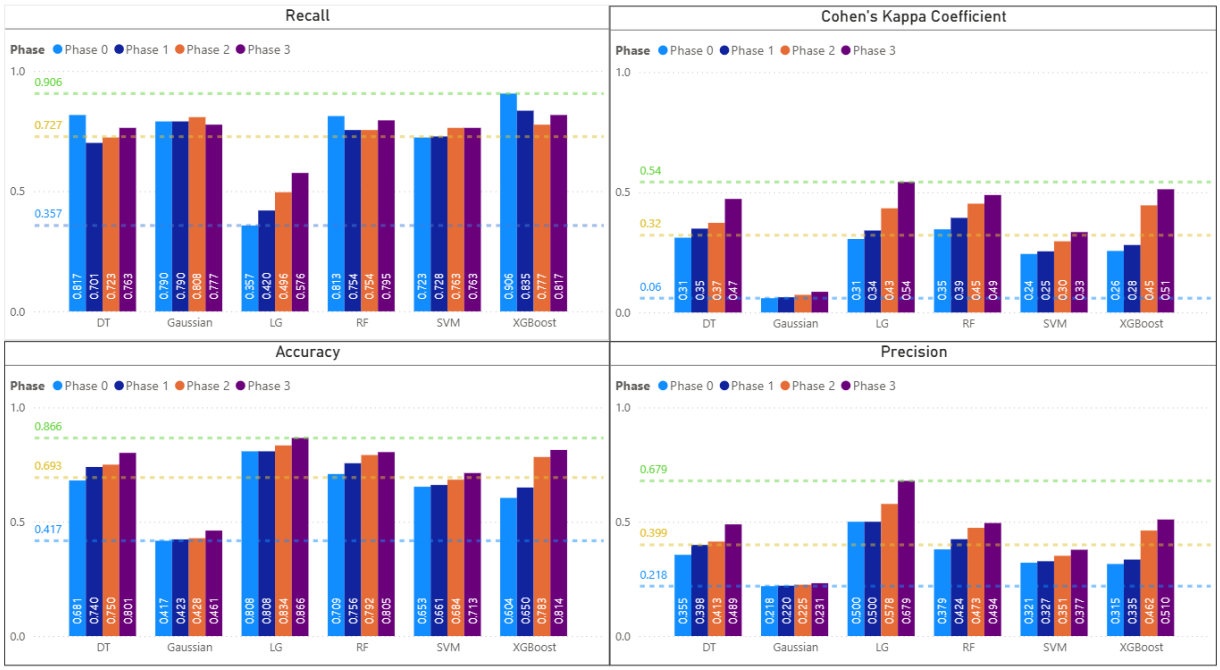


Figure 6.7: Performance comparison of six classifiers (Decision Tree- DT, Gaussian NB, Logistic Regression- LG, Random Forest-RF, SVM, and XGBoost) across four data phases using Recall, Accuracy, F1 Score, and Cohen's Kappa metrics. Phase 0 includes only SIS data, while Phases 1, 2, and 3 integrate both SIS and LMS data. Green, orange, and blue lines represent Max, Average, Min values, respectively.

stages (Phases 0 and 1), the most influential predictors are drawn from SIS data, particularly those reflecting students' academic history. Notably, *Cumm_PreviousTermAtRiskRates* dominates in both Phase 0 (20.52%) and Phase 1 (21.19%), suggesting that prior academic difficulties are strong early indicators of risk. Other key features at this stage include *Student_Term_Number*, *Current_Age*, and *Course_PreviousTerm_AVGDroppedRates*, reflecting the model's dependence on academic maturity and historical course patterns when real-time performance data is limited.

6.5 Conclusion

Using the CRISP-DM framework, this study developed and evaluated predictive models to identify at-risk students early in the academic term. A clear definition of "at-risk" was aligned with institutional policy and derived through in-depth investigation of course registration and outcome records—an often overlooked step in previous studies. Another key difference is how we embedded

Table 6.7: Phase-wise Comparison of RF and XGB Model Performance and Kappa's Agreement

Phase	Key Metrics (RF vs. XGB)	Comment
Phase 0 (Course Start – SIS Only)	Recall: 81.3% vs. 90.6% Accuracy: 70.9% vs. 60.4% Kappa: 0.35 vs. 0.26	XGB achieved higher recall, but RF had stronger accuracy and Kappa. Both fall under "fair agreement," but RF showed greater stability and trustworthiness in this phase.
Phase 1 (30% Course Duration)	Recall: ↓ (RF -5.9%, XGB -7.8%) Accuracy: ↑ (RF +6.6%, XGB +7.6%) Kappa: ↑ (RF +0.04, XGB +0.02)	Early LMS grades introduced noise. Despite slight drops in recall, both models saw improved accuracy and Kappa. RF remained more consistent, although XGB outperformed in Recall.
Phase 2 (50% Course Duration)	Recall:↓ (RF -0%, XGB -5.8%) Accuracy: ↑ (RF +3.6%, XGB +13.3%) Kappa: ↑ (RF +0.06, XGB +0.17)	Additional LMS data led to notable gains.Both RF and XGB's Kappa reached *moderate agreement*; XGB kept outperforming in Recall and almost caught up in Kappa.
Phase 3 (75% Course Duration)	Recall: ↑ (RF +4.1%, XGB +4% Accuracy: ↑ (RF +1.3%, XGB +3.1%) Kappa: ↑ (RF +0.04, XGB +0.06)	With the most complete dataset, XGB outperformed RF across all metrics, achieving moderate agreement and confirming its reliability in later course stages.

the DW development within the data pipelines in the Data Preparation phase. This inclusion allowed us to manage and transform institutional data within a well-defined DW framework. As a result, our design improves both data clarity and reproducibility. When modifications were needed—for instance, we could iteratively revisit and adjust the Data Mart or Analysis Layer without disrupting the upstream pipeline. Additionally, the star-schema structure of our DW enabled easier integration with reporting tools such as Microsoft Power BI for enhanced model interpretation in future.

Among the six evaluated classification algorithms, XGB consistently achieved the highest Recall—ranging from 77% to over 91%—demonstrating its strong ability to detect students at risk, particularly in later phases of the course when more LMS data became available. However, during earlier phases (Phase 0 and Phase 1), where cumulative LMS grading data was sparse or absent, XGB performed less reliably, with lower Accuracy (as low as 60%) and weak Cohen's Kappa scores, indicating predictions close to random. In these early stages, RF provided more stable and trustworthy performance, making it a preferable choice when limited LMS data is available. We also found that different course subjects appeared among the top 10 most important features across phases, likely due to variations in course design—particularly the number and timing of graded items. Additionally, time-aware features had a substantial impact on prediction accuracy. When LMS data was not yet available, SIS-based historical features such as CGPA, earned unit rates, and average course drop rates were among the top predictors. As the course progressed, however, LMS-derived features—including average grades on individual assignments and group-assessment categories—became significantly more influential, improving model performance in later phases. While demographic features such as student age and class size played a relatively minor role (less than 8% importance), they added useful context to the overall prediction.

Feature Name	Phase 0	Phase 1	Phase 2	Phase 3
Cumm_PreviousTermAtRiskRates	20.51%	14.79%	12.14%	7.46%
Course_PreviousTerm_AVGDroppedRates	15.59%	14.09%	10.90%	6.54%
Avg_Avg_GradeGroup_Phase3				28.57%
Class_NoOfStudent	7.90%	7.53%	6.70%	4.66%
Avg_Grade_Phase3				26.11%
Cumm_PreviousTermEarnedUnitRates	6.43%	6.26%	5.25%	3.34%
AVG_PreviousTermNoOfRegisteredUnits	5.73%	5.08%	4.03%	2.47%
Avg_Avg_GradeGroup_Phase2			16.65%	
Avg_Grade_Phase2			14.01%	
Avg_Avg_GradeGroup_Phase1		10.33%		
Avg_Grade_Phase1		9.94%		
Subject_Name_EN_Mathematics	3.68%	2.13%	2.26%	1.81%
Current_Age	3.00%	2.01%	1.96%	1.40%
Student_Term_Number	2.66%		1.77%	1.40%
Total_Units_Term	2.47%	2.25%		
Subject_Name_EN_Engineering	2.11%			

(a) Random Forest (RF)

Feature	Phase 0	Phase 1	Phase 2	Phase 3
Cumm_PreviousTermAtRiskRates	20.52%	21.19%	6.68%	5.26%
Course_PreviousTerm_AVGDroppedRates	9.99%	10.84%	5.37%	3.82%
Student_Term_Number	9.96%	9.61%		
Avg_Grade_Phase3				17.52%
Cumm_PreviousTermEarnedUnitRates	4.92%	5.90%	3.03%	2.66%
Current_Age	7.53%	4.99%		
Total_Units_Term	4.85%	3.41%	2.58%	
Subject_Name_EN_Mechanical Engineering		6.95%	3.82%	
Avg_Avg_GradeGroup_Phase2_Adjusted			9.68%	
Subject_Name_EN_Engineering	5.01%	4.47%		
Avg_Avg_GradeGroup_Phase1_Adjusted		8.83%		
Academic_Load_F			2.94%	4.64%
Subject_Name_EN_Computer Engineering	7.16%			
Subject_Name_EN_English	6.34%			
Subject_Name_EN_Mathematics		6.28%		
Class_NoOfStudent			2.38%	3.65%
Subject_Name_EN_Computer Science	4.17%			
Study_Field_MCGE			3.46%	
Subject_Name_EN_Civil Engineering				3.11%
Class_Language_BI				2.79%
GradingSystem_Weight-based				2.71%
Study_Field_ELGE			2.60%	
Class_Language_FR				2.55%

(b) Gradient Boosting (XGBoost)

Figure 6.8: Top 10 most important features and % important across 4 prediction phases

To extend this work, future research will focus on deploying the model in real learning environments, integrating it into student dashboards for live testing, and collecting feedback to assess its impact on academic engagement. Further improvements may involve incorporating more granular LMS activity data and exploring advanced algorithms such as deep learning to boost predictive power and adaptability.

Chapter 7

Conclusion

This thesis aimed to design, implement, and evaluate a machine learning–based LA model for the early prediction of at-risk undergraduate students at the University of Ottawa. Guided by the DSRM framework, the study addressed four key challenges by formulating and answering four corresponding research questions in a logical sequence. The following section summarizes the major findings and discussions related to each research question, outlines the key contributions of the study within its identified limitations, and proposes directions for future research to build upon this work.

7.1 Summary of Contributions

The DSRM emphasizes the development of both theoretical (conceptual) and practical (implemented and instantiated) artifacts that collectively advance knowledge and demonstrate utility. To ensure coherence between these contributions, each theoretical artifact developed in this study is mapped to its corresponding practical realization. This mapping illustrates how conceptual models, constructs, and methodological frameworks were translated into operational systems and validated within the LA environment at the uOttawa.

Table 7.1 presents the correspondence between the theoretical and practical contributions of this study. Each conceptual artifact developed through the DSRM process was systematically instantiated in a practical implementation to validate its applicability and effectiveness. For example, the conceptual LA infrastructure and layered DW framework were realized through the construction of an operational data warehouse and data pipeline (Practical Contributions 1–2). Similarly, theoretical constructs such as the at-risk student definition and CRISP-DM adaptation were instantiated through machine learning workflows and predictive model development (Practi-

cal Contributions 4–5). This alignment between theoretical design and practical implementation ensures that the research provides both scientific advancement and demonstrable value for institutional decision-making.

7.2 Reflection when Answering Research Questions

7.2.1 How is LA implemented across institutions, and what are the common practices, benefits, and challenges?

Since 2011, LA has grown rapidly worldwide, but institutional-scale implementation in North America—especially Canada—remains limited. For example, the MyLA initiative at the U-M has positively impacted over 6,000 students but remains at the reporting stage, offering dashboards to support student self-reflection. It has not yet advanced to the predictive stage of the LA maturity model, a common limitation in North American institutions.

In Canada, over 50% of HEI have initiated LA efforts, yet most are small-scale, disconnected projects led by individuals rather than coordinated institutional strategies. These implementations also focus primarily on monitoring and reporting rather than predictive or adaptive interventions. While international studies—particularly from the U.S. and Mexico—have shown high predictive accuracy using a range of machine learning models, Canadian contributions to predictive LA remain sparse. A review of 50 papers up to 2021 [10] found no Canadian predictive LA studies, and only 2 of 23 papers from 2019–2023 tested models in real institutional settings.

This gap can be attributed to several challenges. First, integrating data across large, decentralized HEIs requires significant cross-departmental collaboration, technical resources, and funding. Second, the LA development lifecycle involves multiple roles—data engineers, analysts, system architects, and data scientists—making it difficult for small teams to sustain comprehensive efforts. Third, most academic papers focus only on narrow aspects like model prediction, omitting broader architectural or deployment details.

Some Canadian institutions, such as Carleton (Brightspace), York (Moodle), and UoT/UBC/USask (Canvas), have leveraged built-in LMS analytics. However, these remain limited in scope, typically constrained by the LMS's data granularity, course-level focus, and technical barriers to advancing to predictive LA (see Section **??**).

7.2.2 Which Type of Data Infrastructure Is Necessary to Support Scalable LA Implementation at uOttawa?

This study applied the DSRM framework to iteratively design a scalable data infrastructure for LA at uOttawa. While the original goal was to develop a full production-level architecture

Table 7.1: Mapping Between Theoretical and Practical Contributions

Theoretical Contribution	Corresponding Practical Contribution	Outcome / Impact
1.Conceptual LA Infrastructure: A conceptual model integrating data collection, storage, and ML-based analysis for LA processes.	**1. Data Warehouse Implementation:** Development of a DW and data pipeline using a star-schema model and preprocessing framework for ML-based LA.	Established the operational foundation for the uOttawa DataHub for LA.
2. Layered Data Warehouse Framework: A theoretical framework defining layered architecture for educational data organization.	**2. Data Model and Pipeline Development:** Creation of an analytical DW schema and ETL pipeline for data preparation and preprocessing.	Validated feasibility and scalability of the proposed layered DW design.
3. Student Data Construct: List of potential data fields from SIS and Brightspace for LA integration.	**3. Feature Engineering Implementation:** Integration and preprocessing of institutional datasets to generate time-based learning features.	Enabled data standardization and creation of interpretable input variables for ML models.
4. At-Risk Student Definition: New construct aligning academic risk with institutional policy and withdrawal behaviors.	**4. Model Evaluation Framework:** Implementation of ML models to classify students using the new at-risk definition.	Provided a measurable and policy-aligned criterion for risk prediction and validation.
5. Adapted CRISP-DM Methodology: Adaptation of CRISP-DM for educational settings and LA-based ML processes.	**5. ML Workflow Development:** Execution of the adapted CRISP-DM process using Python for exploration, training, validation, and evaluation.	Operationalized CRISP-DM phases within an educational ML context.
6. Phase-to-Phase Predictive Approach: Conceptual design for iterative risk monitoring across the academic term.	**6. Phase-to-Phase Model Implementation:** Development and testing of predictive models at multiple course checkpoints.	Demonstrated continuous risk prediction and early intervention potential.

using Microsoft Azure, institutional constraints—such as restricted data access and limited cloud permissions—necessitated a shift toward a conceptual design. This theoretical artifact lays the groundwork for future institutional deployment.

The proposed architecture follows best practices in data warehousing and aligns with uOttawa's Azure platform. It includes modular layers for staging, orchestration, and analysis, allowing flexible integration of SIS, LMS, and future data sources. The reusable design supports both near-term research and long-term institutional scalability.

However, implementation challenges—such as limited access to uOttawa's systems and the need for cross-departmental coordination—highlight the importance of early planning in large-scale LA initiatives. While not deployed, the architecture contributes a practical blueprint for future efforts and emphasizes the gap between academic design and operational deployment.

7.2.3 What Are the Most Effective ML Approaches for the Early Detection of At-Risk Students Using Empirical Datasets?

Combining the DSRM framework with CRISP-DM, this study developed and evaluated predictive models to identify at-risk students early in the academic term. A clear definition of "at-risk" was aligned with institutional policy and derived through in-depth investigation of course registration and outcome records—an often overlooked step in previous studies.

A key contribution is the integration of DW practices directly into the Data Preparation phase, using the architecture from RQ2. This enabled consistent transformation and flexible reuse of data throughout model development. The star-schema also simplified integration with tools like Power BI for model interpretation.

Modeling addressed class imbalance using SMOTE and class weights, and performance was evaluated using Recall, F1, and Cohen's Kappa. Hyperparameter tuning via `GridSearchCV` with 5-fold cross-validation ensured fairness and reduced overfitting. For example, XGBoost required 135 training rounds across three parameters and five folds.

Despite limited domain-specific hyperparameter knowledge, the models achieved high recall and generalization. Comparative analysis showed that XGB performed best with Brightspace data, especially later in the term, while RF was more reliable in scenarios without LMS activity logs. These findings underscore the importance of aligning model choice with data availability and prediction timing.

7.2.4 What Are the Key Factors Influencing Student Academic Performance?

To answer this question, we conducted model-based feature importance analysis using RF and XGB. Results showed that predictive features vary by course stage, reflecting both data availability and the evolving nature of student engagement.

Early in the term—before Brightspace (LMS) data was available—top predictors came from historical academic records in the SIS. These included prior at-risk rates, earned unit rates, drop rates, term number, and age at registration. These engineered features together contributed over 22% to model importance, emphasizing the role of thoughtful feature design during data preparation.

As courses progressed, LMS-derived features—such as average grades and assignment performance—became dominant. Their rising influence highlights the growing predictive power of real-time engagement data. Demographic features like age and class size, while contributing less than 8%, still offered contextual value.

A key limitation emerged when modeling first-term students who lack historical data. In such cases, imputation (using mean values) maintained model continuity but reduced accuracy, increasing false positives. This trade-off suggests deferring predictions for these students until mid-course (Phase 3), when LMS data improves model reliability. In contrast, for students with academic history, early prediction remains effective.

Overall, this analysis underscores the value of combining dynamic, time-aware features with quality historical data—and the need for tailored imputation strategies to ensure fair predictions across diverse student groups.

7.3 Implication for Research & Practice

7.3.1 Implication for Research

This study offers several contributions to the research community, particularly in the fields of LA and applied ML in education. First, it demonstrates how the CRISP-DM framework can be effectively embedded within DSRM framework to guide the structured development of artifacts—from problem identification through model evaluation. This integrated approach provides a clear methodological pathway for researchers conducting LA projects grounded in institutional needs.

Second, the study underscores the value of using empirical, real-world institutional data rather than pre-cleaned or idealized datasets. By doing so, it highlights the challenges and practical

considerations involved in working with complex academic data, such as missing values, course registration inconsistencies, and data access limitations—issues often overlooked in LA research.

Third, the findings open avenues for future research. Researchers can further optimize or benchmark the current ML approach by experimenting with advanced algorithms (e.g., deep learning, ensemble hybrids) or alternative modeling strategies. Additionally, the study emphasizes the role of feature engineering: future work could investigate the impact of additional or more nuanced features—such as behavioral, demographic, or social variables—on predictive accuracy. This could further enhance early warning systems and contribute to deeper understanding of student success factors.

7.3.2 Implication for Practice

From a practical perspective, the study provides actionable insights for institutions aiming to implement or scale LA initiatives. First, the application of the CRISP-DM framework offers a replicable and transparent approach for developing ML-based LA solutions. Practitioners can adopt this structure to ensure systematic development—from business understanding to deployment—while facilitating interdisciplinary collaboration among data teams.

Second, the proposed DW architecture offers a foundational blueprint for institutions seeking to build scalable, extensible data pipelines for LA. It demonstrates how staging, orchestration, and analysis layers can be designed to support both immediate predictive tasks and longer-term analytics projects. Institutions can use this architecture as a starting point for building centralized data hubs.

Finally, the study's feature importance analysis provides valuable guidance for institutional reporting. By identifying key predictors of student risk—such as cumulative at-risk rates, course drop history, and LMS activity patterns—institutions can design targeted dashboards or intervention strategies. These insights can support academic advisors and learning support staff in making data-informed decisions, enabling more proactive and personalized student support.

7.4 Thesis Limitations

Despite the promising outcomes of this study, several limitations should be acknowledged that may influence the interpretation, generalizability, and future application of its findings.

First, the analysis of the current Learning Analytics (LA) landscape used to address RQ1 and RQ2 relied primarily on publicly accessible institutional documents, including university websites and reports. While these sources provided valuable insights into LA adoption and architectural practices, they are often not peer-reviewed, may change over time, and may lack

technical validation. Nonetheless, due to the limited availability of academic research detailing large-scale LA implementation—particularly in the Canadian context—these public resources served as the most viable and timely option.

Second, the data used for model development and evaluation was drawn from uOttawa's SIS, spanning 2018 to 2023, and from Brightspace LMS, limited to Fall 2024 and courses with substantial LMS adoption. Broader or more recent datasets, such as advising notes, instructor feedback, or real-time student engagement patterns, were not available, which may have limited the depth of analysis and model performance.

Third, the results of this study are institution-specific and may not fully generalize to other educational contexts. Factors such as grading policies, course delivery methods, student demographics, and institutional data structures vary significantly across institutions. While the proposed methodology is transferable, the specific findings—such as feature importance rankings—may differ when applied elsewhere.

Fourth, a particular limitation arises in modeling first-term students. These students lack prior academic history, making it impossible to compute certain key predictive features (e.g., cumulative at-risk rate, previous term GPA). To address this, missing values were imputed using average values from the broader student population. While this strategy allowed the model to remain functional, it likely reduced prediction accuracy for this subgroup. This limitation may partially explain the observed imbalance between recall and precision scores, where the model is prone to over-flagging students with insufficient historical data.

Fifth, due to time and institutional access constraints, the proposed data warehouse and LA architecture could not be fully implemented or tested in a live production environment. Though conceptually complete, deployment would require broader collaboration with university stakeholders and system administrators, as well as extended timelines for integration and validation.

7.5 Future Work

A natural extension of this research is the development and deployment of an early warning system module. Building on the predictive models, data infrastructure, and interpretability framework presented in this thesis, future efforts should focus on:

- Integrating the predictive system into student dashboards used by academic advisors.
- Testing its usability, reliability, and intervention effectiveness in live academic settings.
- Collecting stakeholder feedback (students, advisors, faculty) to refine risk definitions and model outputs.

- Evaluating long-term impact on retention, engagement, and academic performance.

This future deployment would transform the research outcomes into actionable tools for improving student support and institutional decision-making.

Bibliography

[1] U. U. Shaikh and Z. Asif, "Persistence and dropout in higher online education: Review and categorization of factors," *Frontiers in Psychology*, vol. 13, 2022, ISSN: 1664-1078. Accessed: Mar. 12, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpsyg.2022.902070.

[2] D. Olaya, J. Vásquez, S. Maldonado, J. Miranda, and W. Verbeke, "Uplift modeling for preventing student dropout in higher education," *Decision Support Systems*, vol. 134, p. 113 320, Jul. 1, 2020, ISSN: 0167-9236. DOI: 10.1016/j.dss.2020.113320. Accessed: Mar. 12, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923620300750.

[3] C. Carmean, D. Kil, and L. Baer, "Why data matters for student success in a post-pandemic world," Aug. 2021.

[4] T. Irhouma and N. Johnson, "Digital learning in canada in 2022: A changing landscape, 2022 national report," Canadian Digital Learning Research Association, 2022. Accessed: Jan. 21, 2026. [Online]. Available: http://www.cdlra-acrfl.ca/wp-content/uploads/2023/01/2022_national_report_en.pdf.

[5] S. Subiyantoro, "Transformative online learning post-pandemic: Challenges, opportunities, and future trends," *Jurnal Pekommas*, vol. 9, pp. 29–39, Jun. 2024. DOI: 10.56873/jpkm.v9i1.5233.

[6] Studiosity and Angus Reid, *2022 student wellbeing canada*, Accessed: 2024-05-25, 2022. [Online]. Available: https://www.studiosity.com/blog/2022-canadian-student-wellbeing-study-chapter3.

[7] I. Celik, E. Gedrimiene, A. Silvola, et al., "Response of learning analytics to the online education challenges during pandemic: Opportunities and key examples in higher education," *Policy Futures in Education*, 2022. DOI: 10.1177/14782103221078401.

[8] N. Nguyen, N. Tuan, and Y. Takahashi, "Relationship between emotional intelligence and resilience among university students during crisis," *Policy Futures in Education*, vol. 21, p. 147 821 032 211 396, Nov. 2022. DOI: 10.1177/14782103221139620.

[9] J. Mula-Falcón, C. Cruz-González, J. Domingo Segovia, and C. Lucena Rodríguez, "Review of higher education policy during the pandemic: A spanish perspective," *Policy Futures in Education*, vol. 21, no. 4, pp. 465–485, 2022. DOI: 10.1177/14782103221134188.

[10] C. F. de Oliveira, S. R. Sobral, M. J. Ferreira, and F. Moreira, "How does learning analytics contribute to prevent students' dropout in higher education: A systematic literature review," *Big Data and Cognitive Computing*, vol. 5, no. 4, p. 64, Dec. 2021, Number: 4 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2504-2289. DOI: 10.3390/bdcc5040064. Accessed: Feb. 16, 2023. [Online]. Available: https://www.mdpi.com/2504-2289/5/4/64.

[11] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, Dec. 1, 2007, Publisher: Routledge _eprint: https://doi.org/10.2753/MIS0742-1222240302, ISSN: 0742-1222. DOI: 10.2753/MIS0742-1222240302. Accessed: Mar. 1, 2023. [Online]. Available: https://doi.org/10.2753/MIS0742-1222240302.

[12] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The KDD process for extracting useful knowledge from volumes of data," *Communications of the ACM*, vol. 39, no. 11, pp. 27–34, Nov. 1996, ISSN: 0001-0782, 1557-7317. DOI: 10.1145/240455.240464. Accessed: May 5, 2025. [Online]. Available: https://dl.acm.org/doi/10.1145/240455.240464.

[13] F. Martínez-Plumed et al., "CRISP-DM twenty years later: From data mining processes to data science trajectories," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 8, pp. 3048–3061, Aug. 2021, ISSN: 1558-2191. DOI: 10.1109/TKDE.2019.2962680. Accessed: May 5, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/8943998.

[14] C. Schröer, F. Kruse, and J. M. Gómez, "A systematic literature review on applying CRISP-DM process model," *Procedia Computer Science*, CENTERIS 2020 - International Conference on ENTERprise Information Systems / ProjMAN 2020 - International Conference on Project MANagement / HCist 2020 - International Conference on Health and Social Care Information Systems and Technologies 2020, CENTERIS/ProjMAN/HCist 2020, vol. 181, pp. 526–534, Jan. 1, 2021, ISSN: 1877-0509.

DOI: 10.1016/j.procs.2021.01.199. Accessed: May 5, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921002416.

[15] P. Chapman, "CRISP-DM 1.0: Step-by-step data mining guide," 2000. Accessed: May 9, 2025. [Online]. Available: https://www.semanticscholar.org/paper/CRISP-DM-1.0%3A-Step-by-step-data-mining-guide-Chapman/54bad20bbc7938991bf34f86dd

[16] C. Ballard, D. M. Farrell, A. Gupta, C. Mazuela, and S. Vohnik, *Dimensional Modeling: In a Business Intelligence Environment*. Vervante, Feb. 2006, ISBN: 978-0-7384-9644-3.

[17] J. Rodgers and A. Nicewander, "Thirteen ways to look at the correlation coefficient," *American Statistician - AMER STATIST*, vol. 42, pp. 59–66, Feb. 1988. DOI: 10.1080/00031305.1988.10475524.

[18] H. Cramér, *Mathematical Methods of Statistics* (Princeton Mathematical Series). Princeton University Press, 1946, vol. 9.

[19] R. R. Wilcox, "Summarizing data," in *Applying Contemporary Statistical Techniques*, Academic Press, 2003, ch. 3, pp. 55–91, ISBN: 9780127515410. DOI: 10.1016/B978-012751541-0/50024-9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780127515410500249.

[20] M. Feurer and F. Hutter, "Hyperparameter optimization," in May 2019, pp. 3–33, ISBN: 978-3-030-05317-8. DOI: 10.1007/978-3-030-05318-5_1.

[21] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[22] P. Francis, C. Broughan, C. Foster, and C. Wilson, "Thinking critically about learning analytics, student outcomes, and equity of attainment," *Assessment & Evaluation in Higher Education*, vol. 45, no. 6, pp. 811–821, Aug. 17, 2020, Publisher: Routledge _eprint: https://doi.org/10.1080/02602938.2019.1691975, ISSN: 0260-2938. DOI: 10.1080/02602938.2019.1691975. Accessed: Feb. 16, 2023. [Online]. Available: https://doi.org/10.1080/02602938.2019.1691975.

[23] J. A. Larusson and B. White, Eds., *Learning Analytics: From Research to Practice*, New York, NY: Springer New York, 2014, ISBN: 978-1-4614-3304-0 978-1-4614-3305-7. DOI: 10.1007/978-1-4614-3305-7. Accessed: Feb. 7, 2023. [Online]. Available: https://link.springer.com/10.1007/978-1-4614-3305-7.

[24] S. El Alfy, J. Marx Gómez, and A. Dani, "Exploring the benefits and challenges of learning analytics in higher education institutions: A systematic literature review," *Information Discovery and Delivery*, vol. 47, no. 1, pp. 25–34, Jan. 1, 2019, Publisher: Emerald Publishing Limited, ISSN: 2398-6247. DOI: 10.1108/IDD-06-2018-0018. Accessed: Mar. 13, 2023. [Online]. Available: https://doi.org/10.1108/IDD-06-2018-0018.

[25] J. T. Avella, M. Kebritchi, S. G. Nunn, and T. Kanai, "Learning analytics methods, benefits, and challenges in higher education: A systematic literature review," *Online Learning*, vol. 20, no. 2, pp. 13–29, Jun. 2016, Publisher: Online Learning Consortium, Inc ERIC Number: EJ1105911, ISSN: 1939-5256. Accessed: Mar. 13, 2023. [Online]. Available: https://eric.ed.gov/?id=EJ1105911.

[26] N. A. Nurhadi, F. H. Hussin, and M. F. N. Demon, "Predictive learning analytics (PLA) for higher level: A systematic literature review," in *2021 International Conference on Computer & Information Sciences (ICCOINS)*, Jul. 2021, pp. 300–304. DOI: 10.1109/ICCOINS49721.2021.9497170.

[27] O. Khan, "Learners' and teachers' perceptions of learning analytics (LA): A case study of southampton solent university (SSU)," International Association for the Development of the Information Society, Oct. 2017, Publication Title: International Association for Development of the Information Society ERIC Number: ED579477. Accessed: Mar. 13, 2023. [Online]. Available: https://eric.ed.gov/?id=ED579477.

[28] "A bibliometric perspective of learning analytics research landscape," *Behaviour & Information Technology*, 2018. Accessed: Feb. 16, 2023. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/0144929X.2018.1467967.

[29] D. A. Shafiq, M. Marjani, R. A. A. Habeeb, and D. Asirvatham, "Student retention using educational data mining and predictive analytics: A systematic literature review," *IEEE Access*, vol. 10, pp. 72 480–72 503, 2022. DOI: 10.1109/ACCESS.2022.3188767.

[30] J. Zhang, X. Zhang, S. Jiang, P. Ordóñez de Pablos, and Y. Sun, "Mapping the study of learning analytics in higher education," *Behaviour & Information Technology*, vol. 37, no. 10, pp. 1142–1155, Nov. 2, 2018, Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/0144929X.2018.1529198, ISSN: 0144-929X. DOI: 10.1080/0144929X.2018.1529198. Accessed: Feb. 16, 2023. [Online]. Available: https://doi.org/10.1080/0144929X.2018.1529198.

[31] Y. V. Paredes, R. F. Siegle, I.-H. Hsiao, and S. D. Craig, "Educational data mining and learning analytics for improving online learning environments," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 64, no. 1, pp. 500–504, Dec. 1, 2020, Publisher: SAGE Publications Inc, ISSN: 2169-5067. DOI: 10.1177/1071181320641113. Accessed: Feb. 16, 2023. [Online]. Available: https://journals.sagepub.com/doi/abs/10.1177/1071181320641113.

[32] J. Jarke and F. Macgilchrist, "Dashboard stories: How narratives told by predictive analytics reconfigure roles, risk and sociality in education," *Big Data & Society*, vol. 8, no. 1, p. 20 539 517 211 025 561, Jan. 1, 2021, Publisher: SAGE Publications Ltd, ISSN: 2053-9517. DOI: 10.1177/20539517211025561. Accessed: Feb. 27, 2023. [Online]. Available: https://doi.org/10.1177/20539517211025561.

[33] B. Kei Daniel, Ed., *Big Data and Learning Analytics in Higher Education*, Cham: Springer International Publishing, 2017, ISBN: 978-3-319-06519-9 978-3-319-06520-5. DOI: 10.1007/978-3-319-06520-5. Accessed: Mar. 4, 2024. [Online]. Available: http://link.springer.com/10.1007/978-3-319-06520-5.

[34] Q. Li, P. Duffy, and Z. Zhang, "A novel multi-dimensional analysis approach to teaching and learning analytics in higher education," *Systems*, vol. 10, no. 4, p. 96, Aug. 2022, Number: 4 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2079-8954. DOI: 10.3390/systems10040096. Accessed: Feb. 16, 2023. [Online]. Available: https://www.mdpi.com/2079-8954/10/4/96.

[35] "Learning analytics - teaching and learning — university of saskatchewan," Accessed: Feb. 21, 2024. [Online]. Available: https://teaching.usask.ca/documents/gmctl/learning-analytics-scenario2.pdf.

[36] "Data-driven design: Quercus analytics 2021-22 - open UToronto," Accessed: Feb. 21, 2024. [Online]. Available: https://ocw.utoronto.ca/d3qa-21-22/.

[37] A. Cano and J. D. Leonard, "Interpretable multiview early warning system adapted to underrepresented student populations," *IEEE Transactions on Learning Technologies*, vol. 12, no. 2, pp. 198–211, Apr. 2019, Conference Name: IEEE Transactions on Learning Technologies, ISSN: 1939-1382. DOI: 10.1109/TLT.2019.2911079. Accessed: Sep. 30, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/8691619.

[38] C. Guzmán-Valenzuela, C. Gómez-González, A. Rojas-Murphy Tagle, and A. Lorca-Vyhmeister, "Learning analytics in higher education: A preponderance of analytics but very little learning?" *International Journal of Educational Technology in Higher Education*, vol. 18, no. 1, p. 23, 2021, ISSN: 2365-9440. DOI: 10.1186/s41239-021-

00258-x. Accessed: Feb. 21, 2024. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8092999/.

[39] "Show students their data: Using dashboards to support self-regulated learning," EDUCAUSE Review, Accessed: Feb. 20, 2024. [Online]. Available: https://er.educause.edu/articles/2021/7/show-students-their-data-using-dashboards-to-support-self-regulated-learning.

[40] "My learning analytics / u-m information and technology services," Accessed: Nov. 17, 2023. [Online]. Available: https://its.umich.edu/academics-research/teaching-learning/myla-for-students.

[41] "Learning analytics pilots — learning analytics," Accessed: Feb. 20, 2024. [Online]. Available: https://learninganalytics.ubc.ca/about-the-project/tool-pilots/.

[42] U. of Toronto, "Learning analytics initiative report- strategy paper april 2021," 2021. [Online]. Available: https://www.viceprovostundergrad.utoronto.ca/wp-content/uploads/sites/275/2021/07/Revised-Learning-Analytics-Green-Paper.pdf.

[43] "Learning analytics – office of the vice-provost, innovations in undergraduate education," Accessed: Feb. 20, 2024. [Online]. Available: https://www.viceprovostundergrad.utoronto.ca/learning-analytics/.

[44] G. Pate. "Learning analytics and eClass: Improving teaching and learning," Accessed: Feb. 20, 2024. [Online]. Available: https://www.ualberta.ca/the-quad/2017/08/learning-analytics-and-eclass-improving-teaching-and-learning.html.

[45] "Marek hatala homepage, simon fraser university," Accessed: Feb. 20, 2024. [Online]. Available: https://www.sfu.ca/~mhatala/learning-analytics.html.

[46] "Learning analytics - learning technology services," Accessed: Feb. 20, 2024. [Online]. Available: https://lthelp.yorku.ca/learning-analytics.

[47] "Core analytics in brightspace - learning analytics," Accessed: Feb. 20, 2024. [Online]. Available: https://carleton.ca/learninganalytics/learning-analytics-in-brightspace/.

[48] W. University. "Report of the provost's task force for online education june 2020," Accessed: Feb. 20, 2024. [Online]. Available: https://www.provost.uwo.ca/pdf/planning_reports/OTF_final_report_2020June10.pdf.

[49] Q. U. Canada. "Teaching and learning action plan march 2014," Accessed: Feb. 20, 2024. [Online]. Available: https://www.queensu.ca/provost/sites/provwww/files/uploaded_files/TeachingAndLearningActionPlanMarch2014.pdf.

[50] D. Asrani and R. Jain, "Review of techniques used in data warehouse implementation: An initiative towards designing a frame work for effective data warehousing," in *2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014)*, ISSN: 2347-9337, Aug. 2014, pp. 1–5. DOI: 10.1109/ICAETR.2014.7012954.

[51] F. Matsebula and E. Mnkandla, "A big data architecture for learning analytics in higher education," in *2017 IEEE AFRICON*, ISSN: 2153-0033, Sep. 2017, pp. 951–956. DOI: 10.1109/AFRCON.2017.8095610.

[52] P. Kuppusamy and K. Suresh Joseph, "Building an enterprise data lake for educational organizations for prediction analytics using deep learning," in *Proceedings of International Conference on Deep Learning, Computing and Intelligence*, G. Manogaran, A. Shanthini, and G. Vadivu, Eds., ser. Advances in Intelligent Systems and Computing, Singapore: Springer Nature, 2022, pp. 65–81, ISBN: 9789811656521. DOI: 10.1007/978-981-16-5652-1_6.

[53] S. P. M. Choi, S. Lam, K. C. Li, and B. T. M. Wong, "Learning analytics at low cost: At-risk student prediction with clicker data and systematic proactive interventions," *Journal of Educational Technology & Society*, vol. 21, no. 2, pp. 273–290, 2018, Publisher: International Forum of Educational Technology & Society, ISSN: 1176-3647. Accessed: Feb. 17, 2023. [Online]. Available: https://www.jstor.org/stable/26388407.

[54] Google Cloud, *What is a data warehouse?* Accessed: 2023-03-13, 2023. [Online]. Available: https://cloud.google.com/learn/what-is-a-data-warehouse.

[55] G. Siemens and P. Long, "Penetrating the fog: Analytics in learning and education," *EDUCAUSE Review*, 2011.

[56] H. Coates, "The value of student engagement for higher education quality assurance," *Quality in Higher Education*, 2005.

[57] "Successfully migrating from blackboard to brightspace," D2L, Accessed: Feb. 25, 2024. [Online]. Available: https://www.d2l.com/why-d2l/customers/university-of-ottawa/.

[58] "Understanding how student success system works," Brightspace, Accessed: Nov. 17, 2023. [Online]. Available: https://community.d2l.com/brightspace/kb/articles/5657-understanding-how-student-success-system-works.

[59] B. Albreiki, N. Zaki, and H. Alashwal, "A systematic literature review of student' performance prediction using machine learning techniques," *Education Sciences*, vol. 11, no. 9, pp. 1–27, 2021. DOI: 10.3390/educsci11090552. [Online]. Available: https://doi.org/10.3390/educsci11090552.

[60] A. Jokhan, A. Chand, V. Singh, and K. Mamun, "Increased digital resource consumption in higher educational institutions and the artificial intelligence role in informing decisions related to student performance," *Sustainability (Switzerland)*, vol. 14, no. 4, 2022, ISSN: 2071-1050. DOI: 10.3390/su14042377.

[61] F. Xiang, X. Zhang, J. Cui, M. Carlin, and Y. Song, "Algorithmic bias in a student success prediction models: Two case studies," in *2022 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, ISSN: 2470-6698, Dec. 2022, pp. 310–315. DOI: 10.1109/TALE54877.2022.00058. Accessed: Sep. 30, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10148378.

[62] C. Kaensar and W. Wongnin, "Analysis and prediction of student performance based on moodle log data using machine learning techniques," *International Journal of Emerging Technologies in Learning*, vol. 18, no. 10, pp. 184–203, 2023, ISSN: 1868-8799. DOI: 10.3991/ijet.v18i10.35841.

[63] A. Esteban, C. Romero, and A. Zafra, "Assignments as influential factor to improve the prediction of student performance in online courses," *Applied Sciences (Switzerland)*, vol. 11, no. 21, 2021, ISSN: 2076-3417. DOI: 10.3390/app112110145.

[64] V. Anderková, T. Adam, and F. Babič, "Data-driven student performance prediction," in *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Mar. 2022, pp. 000 267–000 272. DOI: 10.1109/SAMI54271.2022.9780854. Accessed: Sep. 30, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9780854.

[65] A. K. Dileep, A. Bansal, and J. Cunningham, "Early detection of at-risk students in a calculus course," in *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, ISSN: 0730-3157, Jun. 2022, pp. 187–194. DOI: 10.1109/COMPSAC54236.2022.00034. Accessed: Sep. 30, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9842632.

[66] H. Park and S. Yoo, "Early dropout prediction in online learning of university using machine learning," *International Journal on Informatics Visualization*, vol. 5, no. 4, pp. 347–353, 2021, ISSN: 2549-9904. DOI: 10.30630/JOIV.5.4.732.

[67] E. Alhazmi and A. Sheneamer, "Early predicting of students performance in higher education," *IEEE Access*, vol. 11, pp. 27 579–27 589, 2023, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2023.3250702. Accessed: Sep. 30, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10056943/.

[68] H. Wan, M. Li, Z. Zhong, and X. Luo, "Early prediction of student performance with LSTM-based deep neural network," in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, Torino, Italy: IEEE, Jun. 2023, pp. 132–141, ISBN: 9798350326970. DOI: 10.1109/COMPSAC57700.2023.00026. Accessed: Oct. 10, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10196967/.

[69] S. Ghashout, Y. Gdura, and N. Drawil, "Early prediction of students' academic performance using artificial neural network: A case study in computer engineering department," in *2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, May 2023, pp. 40–45. DOI: 10.1109/MI-STA57575.2023.10169421. Accessed: Sep. 30, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10169421.

[70] S. Gaftandzhieva et al., "Exploring online activities to predict the final grade of student," *Mathematics*, vol. 10, no. 20, 2022, ISSN: 2227-7390. DOI: 10.3390/math10203758.

[71] H. Turabieh, "Hybrid machine learning classifiers to predict student performance," in *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, Amman, Jordan: IEEE, Oct. 2019, pp. 1–6, ISBN: 978-1-72812-882-5. DOI: 10.1109/ICTCS.2019.8923093. Accessed: Sep. 30, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/8923093/.

[72] S. Bai, D. Zhu, and W. Dang, "Research on the prediction model of the college student performance based on the multi-attention mechanism," in *Fourth International Conference on Computer Science and Educational Informatization (CSEI 2022)*, vol. 2022, Sep. 2022, pp. 78–85. DOI: 10.1049/icp.2022.1457. Accessed: Sep. 30, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9957037.

[73] B. M. Neda, M. Wang, A. Singh, S. Gago-Masague, and J. Wong-Ma, "Staying ahead of the curve: Early prediction of academic probation among first-year CS students," in *2023 3rd International Conference on Applied Artificial Intelligence (ICAPAI)*, Halden, Norway: IEEE, May 2, 2023, pp. 1–7, ISBN: 9798350328929. DOI: 10.1109/ICAPAI58366.2023.10194020. Accessed: Sep. 30, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10194020/.

[74] Y. Joshi, K. Mallibhat, and V. M., "Students' performance prediction using multimodal machine learning," in *2022 IEEE IFEES World Engineering Education Forum - Global Engineering Deans Council (WEEF-GEDC)*, Nov. 2022, pp. 1–5. DOI: 10.1109/WEEF-GEDC54384.2022.9996212. Accessed: Sep. 30, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9996212.

[75] R. Bertolini, S. J. Finch, and R. H. Nehm, "Testing the impact of novel assessment sources and machine learning methods on predictive outcome modeling in undergraduate biology," *Journal of Science Education and Technology*, vol. 30, no. 2, pp. 193–209, Apr. 1, 2021, ISSN: 1573-1839. DOI: 10.1007/s10956-020-09888-8. Accessed: Sep. 30, 2023. [Online]. Available: https://doi.org/10.1007/s10956-020-09888-8.

[76] I. Khan, A. Al Sadiri, A. R. Ahmad, and N. Jabeur, "Tracking student performance in introductory programming by means of machine learning," in *2019 4th MEC International Conference on Big Data and Smart City (ICBDSC)*, Jan. 2019, pp. 1–6. DOI: 10.1109/ICBDSC.2019.8645608. Accessed: Sep. 30, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/8645608.

[77] H. A. Mengash, "Using data mining techniques to predict student performance to support decision making in university admission systems," *IEEE Access*, vol. 8, pp. 55 462–55 470, 2020, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2981905. Accessed: Sep. 30, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9042216?denied=.

[78] M. Adnan, A. A. S. Alarood, M. I. Uddin, and I. u. Rehman, "Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models," *PeerJ Computer Science*, vol. 8, e803, Feb. 21, 2022, Publisher: PeerJ Inc., ISSN: 2376-5992. DOI: 10.7717/peerj-cs.803. Accessed: Sep. 30, 2023. [Online]. Available: https://peerj.com/articles/cs-803.

[79] M. Khan, S. Naz, Y. Khan, M. Zafar, M. Khan, and G. Pau, "Utilizing machine learning models to predict student performance from LMS activity logs," *IEEE Access*, vol. 11, pp. 86 953–86 962, 2023, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2023.3305276. Accessed: Sep. 30, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10216987/.

[80] Y. M. I. Hassan, A. Elkorany, and K. Wassif, "Utilizing social clustering-based regression model for predicting student's GPA," *IEEE Access*, vol. 10, pp. 48 948–48 963, 2022, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3172438. Accessed: Sep. 30, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9767814.

[81] Y. Chen and L. Zhai, "XGBoost-based student performance prediction in tiered instruction," in *Fourth International Conference on Computer Science and Educational Informatization (CSEI 2022)*, Hybrid Conference, Taiyuan, China: Institution of Engineering and Technology, 2022, pp. 59–64, ISBN: 978-1-83953-788-2. DOI: 10.1049/icp.2022.1454. Accessed: Sep. 30, 2023. [Online]. Available: https://digital-library.theiet.org/content/conferences/10.1049/icp.2022.1454.

[82] T. Lam and A. Sowinski, "The current canadian landscape on learning analytics," *Proceedings of the Canadian Engineering Education Association (CEEA)*, Dec. 2024. DOI: 10.24908/pceea.2024.18604. [Online]. Available: https://ojs.library.queensu.ca/index.php/PCEEA/article/view/18604.

[83] U. of Ottawa, *Important academic dates and deadlines*, Accessed: 2024-05-23, 2024. [Online]. Available: https://www.uottawa.ca/study/important-academic-dates-deadlines.

[84] U. of Ottawa, *Grading system*, Accessed: 2024-05-23, 2024. [Online]. Available: https://www.uottawa.ca/about-us/leadership-governance/policies-regulations/a-3-grading-system.

[85] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[86] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1–2, pp. 273–324, 1997. DOI: 10.1016/S0004-3702(97)00043-X.

[87] C. Romero and S. Ventura, "Educational data mining: A review of the state of the art," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 6,

[88] K. Pearson, "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 5th ser., vol. 50, no. 302, pp. 157–175, 1900. DOI: 10.1080/14786440009463897.

[89] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 785–794, ISBN: 9781450342322. DOI: 10.1145/2939672.2939785. [Online]. Available: https://doi.org/10.1145/2939672.2939785.

[90] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," *Encyclopedia of Database Systems*, vol. 532–538, pp. 532–538, Jan. 2009. DOI: 10.1007/978-0-387-39940-9_565.

[91] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res. (JAIR)*, vol. 16, pp. 321–357, Jun. 2002. DOI: 10.1613/jair.953.

[92] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960. DOI: 10.1177/001316446002000104. [Online]. Available: https://doi.org/10.1177/001316446002000104.

[93] P. Czodrowski, "Count on kappa," *Journal of Computer-Aided Molecular Design*, vol. 28, no. 11, pp. 1049–1055, Nov. 1, 2014, ISSN: 1573-4951. DOI: 10.1007/s10822-014-9759-6. [Online]. Available: https://doi.org/10.1007/s10822-014-9759-6.

[94] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977, ISSN: 0006341X, 15410420. Accessed: May 22, 2025. [Online]. Available: http://www.jstor.org/stable/2529310.

APPENDICES

Appendix A

Proposed Student-Related Data Categories in uOttawa

Table A.1: Overview of Student-Related Data Features and Availability Status at the Time of Request

Group	Feature Group	Feature	Available
Personal Data	Demographic	Gender	Yes
		Age	Yes
		Citizen status	Yes
		Visa status	Yes
		Country of visa/citizen	Yes
		Original country	Calculated
		Country of residency	Yes
Admission Data	Entry / Qualification	Program start date	Yes
		Program effective date	Yes
		Graduate degree	Yes
		Institution name	Yes
		Program name	Yes
		Faculty	Yes
		Program career	Yes
		Program language	Yes

Continued on next page

Table A.1 continued

Group	Feature Group	Feature	Available
		Calculated GPA	Calculated
Admission Data	Background	Degree name	No
		Origin education institution	No
		Province of origin institution	No
		Country of origin institution	No
		Is transferred?	No
		Transfer type	No
Academic Activities	Online Activity (Brightspace)	Login times	No
		Course content page visits	No
		Quiz attempts	No
		Discussion forum views	No
		Original forum posts	No
		Forum replies	No
		Content downloads	No
	Enrolled Course Records	Enrolled course name	Yes
		Enrolled date	Yes
		Dropped date	Yes
		Course subject	yes
		Course modality	Yes
		Course language	Yes
		Course grade	No
		Assignment grades	Yes
		Assignment date	Yes
		Course syllabus	No
	Microsoft Teams Meetings	Login event records	No
		Message records	No
	Help	Tutor / Library support	No
Academic Performance	Course Results	Course name	Yes
		Degree name	Yes
		GPA	Calculated
		Academic year	Yes

Continued on next page

Table A.1 continued

Group	**Feature Group**	**Feature**	**Available**
		Course attempts	Calculated
		Assessment	No
		Is drop out?	Calculated
		Graduation results	No
		Result date	No
Others	Financial Aid	Academic year	No
		Scholarship type	No
		Scholarship name	No
		Student loan / grant	No
	Supports	–	No
	Campus information	–	No
	Joined clubs	–	No
	Joined events / seminars	–	No
	WiFi usage	–	No

Table A.2: Brightspace Data Sources and Availability Status at the Time of Request

No	Dataset Type	Dataset Name	Description	Available
1	Advanced Dataset	Content Progress	Content Progress provides insight into content viewed and not viewed by users across selected organizational units within a specified date range.	No
2	Advanced Dataset	Learner Usage	User engagement information.	No
3	Brightspace Dataset	Assignments Data Sets	Assignment-related information generated during learning activities.	No
4	Brightspace Dataset	Content Data Sets	Content-level information reflecting student engagement with posted learning materials.	No
5	Brightspace Dataset	Grades Data Sets	Grade information associated with assignments.	Yes
6	Brightspace Dataset	Quizzes Data Sets	Information related to quizzes and assessment activities.	No
7	Brightspace Dataset	Users Data Sets	User-level engagement information.	No

Appendix B

Developing Data Warehouse and Data Pipelines- SQL Scriptings

B.0.1 DW Layer-Developing Dimension Tables

B.0.1.1 Dim_ProgramPlan

```
CREATE TABLE [dbo].[Dim_ProgramPlan](
        [Dim_ProgramPlan_Key] [int] IDENTITY(1,1) NOT NULL
         ,Institution_Key int   NULL
        , Faculty_Service_Key  int NULL
        , Academic_Career_Key int   NULL
        , Academic_Program_Key int  NULL
        , Academic_Plan_Key int  NULL
        ,Degree_Key int  NULL
        ,Institution_Name_EN nvarchar(max) NULL
        ,Faculty_Service_Name_EN nvarchar(max) NULL
        ,Faculty_Service_Name_FR nvarchar(max) NULL
        , Faculty_Service_Type nvarchar(max) NULL
        , Faculty_Service_Name_Short_EN nvarchar(max) NULL
        , Faculty_Service_Name_Short_FR nvarchar(max) NULL
        , Academic_Career_Code nvarchar(max) NULL
        , Academic_Career_Name_FR nvarchar(max) NULL
        ,Academic_Career_Name_EN nvarchar(max) NULL
        ,Program_Name_EN nvarchar(max) NULL
        ,Program_Name_FR nvarchar(max) NULL
```

```
        ,Program_Name_Long_EN nvarchar(max) NULL
        ,Program_Name_Long_FR nvarchar(max) NULL
        ,Program_Credits_Required int  NULL
        ,Program_Dual nvarchar(max) NULL
        ,Program_Normal_Completion int  NULL
        ,uoCampus_Program_Code nvarchar(max) NULL
        ,Academic_Plan_Name_FR nvarchar(max) NULL
        ,Academic_Plan_Name_EN nvarchar(max) NULL
        ,Academic_Plan_Name_Long_EN nvarchar(max) NULL
        ,Academic_Plan_Name_Long_FR nvarchar(max) NULL
        ,Academic_Plan_Type nvarchar(max) NULL
        ,Diploma_Description_EN nvarchar(max) NULL
        ,Diploma_Description_FR nvarchar(max) NULL
        ,Academic_Plan_Language nvarchar(max) NULL
        ,Study_Field nvarchar(max) NULL
        ,uoCampus_Plan_Code nvarchar(max) NULL

 CONSTRAINT [PK_Dim_ProgramPlan] PRIMARY KEY CLUSTERED
(
        [Dim_ProgramPlan_Key] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
   OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

Listing B.1: Table Creation Scripting for dimension table: Dim_ProgramPlan

```

---INSERT DATA TO DBO.DIM_PROGRAMPLAN---
--Use the following truncate statement when reloading the table
--Truncate table [dbo].[Dim_ProgramPlan];
insert into [dbo].[Dim_ProgramPlan] (
        Institution_Key
        , Faculty_Service_Key
        , Academic_Career_Key
        , Academic_Program_Key
        , Academic_Plan_Key
        ,Degree_Key
        ,Institution_Name_EN
        ,Faculty_Service_Name_EN
        ,Faculty_Service_Name_FR
```

```
        , Faculty_Service_Type
        , Faculty_Service_Name_Short_EN
        , Faculty_Service_Name_Short_FR
        , Academic_Career_Code
        , Academic_Career_Name_FR
        ,Academic_Career_Name_EN
        ,Program_Name_EN
        ,Program_Name_FR
        ,Program_Name_Long_EN
        ,Program_Name_Long_FR
        ,Program_Credits_Required
        ,Program_Dual
        , Program_Normal_Completion
        ,uoCampus_Program_Code
        ,Academic_Plan_Name_FR
        ,Academic_Plan_Name_EN
        ,Academic_Plan_Name_Long_EN
        ,Academic_Plan_Name_Long_FR
        ,Academic_Plan_Type
        ,Diploma_Description_EN
        ,Diploma_Description_FR
        ,Academic_Plan_Language
        ,Study_Field
        ,uoCampus_Plan_Code)

--Data Retrieval Statement -----
select
        A.Institution_Key
        , B.Faculty_Service_Key
        , C.Academic_Career_Key
        , E.Academic_Program_Key
        , F.Academic_Plan_Key
        ,F.Degree_Key
        ,A.Institution_Name_EN
        ,B.Faculty_Service_Name_EN
        ,B.Faculty_Service_Name_FR
        , B.Faculty_Service_Type
        , B.Faculty_Service_Name_Short_EN
        , B.Faculty_Service_Name_Short_FR
        , C.Academic_Career_Code
```

```
55          , C.Academic_Career_Name_FR
56          ,C.Academic_Career_Name_EN
57          ,E.Program_Name_EN
58          ,E.Program_Name_FR
59          ,E.Program_Name_Long_EN
60          ,E.Program_Name_Long_FR
61          ,E.Program_Credits_Required
62          ,E.Program_Dual
63          ,E.Program_Normal_Completion
64          ,E.uoCampus_Program_Code
65          ,F.Academic_Plan_Name_FR
66          ,F.Academic_Plan_Name_EN
67          ,F.Academic_Plan_Name_Long_EN
68          ,F.Academic_Plan_Name_Long_FR
69          ,F.Academic_Plan_Type
70          ,F.Diploma_Description_EN
71          ,F.Diploma_Description_FR
72          ,F.Academic_Plan_Language
73          ,F.Study_Field
74          ,F.uoCampus_Plan_Code
75
76  from ((([common].[Institution_d] A
77          right join [common].[Faculty_Service_d] B on (A.
               Institution_Key=B.Institution_Key))
78          right join [uocampus].[Academic_Career_d] C on (A.
               Institution_Key=C.Institution_Key))
79          right join [uocampus].[Academic_Program_d] E on (A.
               Institution_Key=E.Institution_Key and B.
               Faculty_Service_Key=E.Faculty_Service_Key and C.
               Academic_Career_Key=E.Program_Academic_Career_Key))
80          right join [uocampus].[Academic_Plan_d] F on ( E.
               Academic_Program_Key=F.Academic_Program_Key)
81  Where F.Academic_Plan_Status='A' and C.[Academic_Career_Status]='A
       ' and E.[Program_Status]='A'
82  order by A.Institution_Key
83          , B.Faculty_Service_Key
84          , C.Academic_Career_Key
85          , E.Academic_Program_Key
86          , F.Academic_Plan_Key
```

Listing B.2: Inserting and Retrieving Scripting for dimension table: Dim_ProgramPlan

B.0.1.2 Dim_Program

```
CREATE TABLE [dbo].[Dim_Program](
        [Dim_Program_Key] [int] IDENTITY(1,1) NOT NULL,
        [Institution_Key] [int] NULL,
        [Faculty_Service_Key] [int] NULL,
        [Academic_Career_Key] [int] NULL,
        [Academic_Program_Key] [int] NULL,
        --[Academic_Plan_Key] [int] NULL,
        --[Degree_Key] [int] NULL,
        [Institution_Name_EN] [nvarchar](max) NULL,
        [Faculty_Service_Name_EN] [nvarchar](max) NULL,
        [Faculty_Service_Name_FR] [nvarchar](max) NULL,
        [Faculty_Service_Type] [nvarchar](max) NULL,
        [Faculty_Service_Name_Short_EN] [nvarchar](max) NULL,
        [Faculty_Service_Name_Short_FR] [nvarchar](max) NULL,
        [Academic_Career_Code] [nvarchar](max) NULL,
        [Academic_Career_Name_FR] [nvarchar](max) NULL,
        [Academic_Career_Name_EN] [nvarchar](max) NULL,
        [Program_Name_EN] [nvarchar](max) NULL,
        [Program_Name_FR] [nvarchar](max) NULL,
        [Program_Name_Long_EN] [nvarchar](max) NULL,
        [Program_Name_Long_FR] [nvarchar](max) NULL,
        [Program_Credits_Required] [int] NULL,
        [Program_Dual] [nvarchar](max) NULL,
        [Program_Normal_Completion] [int] NULL,
        [uoCampus_Program_Code] [nvarchar](max) NULL,
        --[Academic_Plan_Name_FR] [nvarchar](max) NULL,
        --[Academic_Plan_Name_EN] [nvarchar](max) NULL,
        --[Academic_Plan_Name_Long_EN] [nvarchar](max) NULL,
        --[Academic_Plan_Name_Long_FR] [nvarchar](max) NULL,
        --[Academic_Plan_Type] [nvarchar](max) NULL,
        --[Diploma_Description_EN] [nvarchar](max) NULL,
        --[Diploma_Description_FR] [nvarchar](max) NULL,
        --[Academic_Plan_Language] [nvarchar](max) NULL,
```

```
        --[Study_Field] [nvarchar](max) NULL,
        --[uoCampus_Plan_Code] [nvarchar](max) NULL,
 CONSTRAINT [PK_Dim_Program] PRIMARY KEY CLUSTERED
(
        [Dim_Program_Key] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
   OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

Listing B.3: Table Creation Scripting for dimension table: Dim_Program

–Select Statement to insert —– select A.Institution$_{K}ey, B.Faculty_{S}ervice_{K}ey, C.Academic_{C}areer_{K}ey, E.Ac$
$-, F.Academic_{P}lan_{K}ey--, F.Degree_{K}ey, A.Institution_{N}ame_{E}N, B.Faculty_{S}ervice_{N}ame_{E}N, B.Faculty_{S}ervi$
$-, F.Academic_{P}lan_{N}ame_{F}R--, F.Academic_{P}lan_{N}ame_{E}N--, F.Academic_{P}lan_{N}ame_{L}ong_{E}N-$
$-, F.Academic_{P}lan_{N}ame_{L}ong_{F}R--, F.Academic_{P}lan_{T}ype--, F.Diploma_{D}escription_{E}N-$
$-, F.Diploma_{D}escription_{F}R--, F.Academic_{P}lan_{L}anguage--, F.Study_{F}ield--, F.uoCampus_{P}lan_{C}ode$

from ((([common].[Institution$_{d}]Arightjoin[common].[Faculty_{S}ervice_{d}]Bon(A.Institution_{K}ey =$
$B.Institution_{K}ey))rightjoin[uocampus].[Academic_{C}areer_{d}]Con(A.Institution_{K}ey = C.Institution_{K}ey))right$
$E.Institution_{K}eyandB.Faculty_{S}ervice_{K}ey = E.Faculty_{S}ervice_{K}eyandC.Academic_{C}areer_{K}ey =$
$E.Program_{A}cademic_{C}areer_{K}ey))--rightjoin[uocampus].[Academic_{P}lan_{d}]Fon(E.Academic_{P}rogram_{K}ey =$
$F.Academic_{P}rogram_{K}ey)Where--F.Academic_{P}lan_{S}tatus =' A'andC.[Academic_{C}areer_{S}tatus] ='$
$A'andE.[Program_{S}tatus] =' A'orderbyA.Institution_{K}ey, B.Faculty_{S}ervice_{K}ey, C.Academic_{C}areer_{K}ey, E.Ac$
$-, F.Academic_{P}lan_{K}ey$

B.0.1.3 Dim_CourseInfo

```
CREATE TABLE dbo.[Dim_CourseInfo](
                [Dim_CourseInfo_Key] [int] IDENTITY(1,1) NOT NULL,
                [Institution_Key][int] NOT NULL,
                Faculty_Service_Key [int] NULL,
                Academic_Career_Key [int] NULL,
                Academic_Unit_Key [int] NULL,
                Subject_Key [int] NULL,
                Course_Key [int] NOT NULL,
                Faculty_Service_Name_EN [nvarchar](max) NULL,
                Faculty_Service_Name_FR [nvarchar](max) NULL,
                Academic_Career_Name_EN  [nvarchar](max) NULL,
                Academic_Career_Name_FR  [nvarchar](max) NULL,
                Academic_Unit_Name_EN [nvarchar](max) NULL,
```

```
                Academic_Unit_Name_FR [nvarchar](max) NULL,
                Subject_Name_EN [nvarchar](max) NULL,
                Subject_Name_FR [nvarchar](max) NULL,
                Subject_Code [nvarchar](max) NULL,
                Catalog_Number [nvarchar](max) NULL,
                Course_Name_EN [nvarchar](max) NULL,
                Course_Name_FR [nvarchar](max) NULL,
                [uoCampus_Course_ID] [nvarchar](max) NULL,
                [Course_Effective_Date]  [int] NOT NULL,
                [Course_Units] [decimal](4, 2) NULL,
                CourseCode [nvarchar](max) NULL,

 CONSTRAINT [PK_Dim_CourseInfo] PRIMARY KEY CLUSTERED
(
        [Dim_CourseInfo_Key] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
   OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

Listing B.4: Creation Scripting for dimension table: Dim_CourseInfo

```
--Use the following truncate statement when reloading the table
--truncate table Dim_CourseInfo;
insert into Dim_CourseInfo (
                [Institution_Key]
                ,Faculty_Service_Key
                , Academic_Career_Key
                , Academic_Unit_Key
                , Subject_Key
                , Course_Key
                , Faculty_Service_Name_EN
                , Faculty_Service_Name_FR
                , Academic_Career_Name_EN
                , Academic_Career_Name_FR
                , Academic_Unit_Name_EN
                , Academic_Unit_Name_FR
                , Subject_Name_EN
                , Subject_Name_FR
                , Subject_Code
                , Catalog_Number
```

```
                , Course_Name_EN
                , Course_Name_FR
                , [uoCampus_Course_ID]
                , [Course_Effective_Date]
                , [Course_Units]
                , CourseCode)
--Data Retrieval Statement
select  D.[Institution_Key]
                , C.Faculty_Service_Key
                , E.Academic_Career_Key
                , B.Academic_Unit_Key
                , A.Subject_Key
                , D.Course_Key
                , C.Faculty_Service_Name_EN
                , C.Faculty_Service_Name_FR
                , E.Academic_Career_Name_EN
                , E.Academic_Career_Name_FR
                , B.Academic_Unit_Name_EN
                , B.Academic_Unit_Name_FR
                , A.Subject_Name_EN
                , A.Subject_Name_FR
                , A.Subject_Code
                , D.Catalog_Number
                , D.Course_Name_EN
                , D.Course_Name_FR
                , D.[uoCampus_Course_ID]
                , D.[Course_Effective_Date]
                , D.[Course_Units]
                ,A.Subject_Code + D.Catalog_Number as CourseCode


from
        (
         [uocampus].[Course_d] D
         left join [uocampus].[Academic_Career_d] E on (D.
            Course_Academic_Career_Key=E.Academic_Career_Key))
        left join [uocampus].[Subject_d] A  on (A.Subject_Key=D.
           Subject_Key)

        left join [uocampus].[Academic_Unit_d] B on (A.
```

```
            Academic_Unit_Key=B.Academic_Unit_Key)
        left join [common].[Faculty_Service_d] C on (B.
            Faculty_Service_Key=C.Faculty_Service_Key )


where [Subject_Status]='A'
        order by  C.[Institution_Key]
                , C.Faculty_Service_Key
                , E.Academic_Career_Key
                , B.Academic_Unit_Key
                , A.Subject_Key
                , D.Course_Key"
```

Listing B.5: Inserting and Retrieving Scripting for dimension table: Dim_CourseInfo

B.0.1.4 Dim_AcademicTermCode

```
CREATE TABLE [dbo].[Dim_AcademicTermCode](
        [Academic_Year] [nvarchar](max) NULL,
        [Academic_Year_Name] [nvarchar](max) NULL,
        [Academic_Year_Start_Date] [date] NULL,
        [Academic_Term_Code] [int] Not NULL,
        [Academic_Term_EN] [nvarchar](max) NULL,
        [Academic_Term_FR] [nvarchar](max) NULL,
        [Academic_Term_Year_EN] [nvarchar](max) NULL,
        [Academic_Term_Year_FR] [nvarchar](max) NULL,
        [External_Academic_Year] [nvarchar](max) NULL,
        [External_Academic_Year_Start_Date] [date] NULL,

 CONSTRAINT [PK_Academic_Term_Code] PRIMARY KEY CLUSTERED
(
        [Academic_Term_Code] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
   OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

Listing B.6: Creation Scripting for dimension table: Dim_AcademicTermCode

```
--Use the following truncate statement when reloading the table
--truncate table dbo.[Dim_AcademicTermCode];
insert into dbo.[Dim_AcademicTermCode] ([Academic_Year]
      ,[Academic_Year_Name]
      ,[Academic_Year_Start_Date]
      ,[Academic_Term_Code]
      ,[Academic_Term_EN]
      ,[Academic_Term_FR]
      ,[Academic_Term_Year_EN]
      ,[Academic_Term_Year_FR]
      ,[External_Academic_Year]
      ,[External_Academic_Year_Start_Date])

--Select Statement to insert
SELECT Distinct
      [Academic_Year]
      ,[Academic_Year_Name]
      ,[Academic_Year_Start_Date]
      ,[Academic_Term_Code]
      ,[Academic_Term_EN]
      ,[Academic_Term_FR]
      ,[Academic_Term_Year_EN]
      ,[Academic_Term_Year_FR]
      ,[External_Academic_Year]
      ,[External_Academic_Year_Start_Date]

  FROM [common].[Date_d]
  where Academic_Year_Start_Date >= '2000-05-01'
  order by [Academic_Term_Code]
```

Listing B.7: Inserting and Retrieving Scripting for dimension table: Dim_AcademicTermCode

B.0.1.5 Dim_ClassTerm

```
CREATE TABLE dbo.[Dim_ClassTerm](
        [Dim_ClassEnrollment_Key] [int] IDENTITY(1,1) NOT NULL,
        [Institution_Key] [int] NULL
        ,Faculty_Service_Key [int]NULL
```

```
                    ,Academic_Career_Key[int] NULL
                    , Academic_Unit_Key [int] NULL
                    , Subject_Key [int] NULL
                    , Course_Key [int] NULL
                    ,Academic_Term_Code [int] NULL
                    ,Modality_Key [int] NULL
                    ,Class_Term_Key [int] NOT NULL
                    ,Session_Code [nvarchar](10) NULL
                    , Faculty_Service_Name_EN [nvarchar](max) NULL
                    , Faculty_Service_Name_FR [nvarchar](max) NULL
                    , Academic_Career_Name_EN  [nvarchar](max) NULL
                    , Academic_Career_Name_FR  [nvarchar](max) NULL
                    , Academic_Unit_Name_EN [nvarchar](max) NULL
                    , Academic_Unit_Name_FR [nvarchar](max) NULL
                    , Subject_Name_EN [nvarchar](max) NULL
                    , Subject_Name_FR [nvarchar](max) NULL
                    , Course_Code  [nvarchar](max) NULL
                    , Course_Name_EN [nvarchar](max) NULL
                    , Course_Name_FR [nvarchar](max) NULL
                    ,uoCampus_Class_Nbr [nvarchar](max) NULL
                    ,Class_Start_Date [int] NULL
                    ,Class_End_Date [int] NULL
                    ,Class_SSR_Component [nvarchar](10) NULL
                    ,Class_Cancel_Date [int] NULL
                    ,Class_Section [nvarchar](10) NULL
                    ,Class_Language [nvarchar](max) NULL
                    ,Instruction_Mode [nvarchar](max) NULL
                    ,Modality_Description [nvarchar](max) NULL,

 CONSTRAINT [PK_Dim_ClassEnrollment] PRIMARY KEY CLUSTERED
(
        [Dim_ClassEnrollment_Key] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
   OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

Listing B.8: Creation Scripting for dimension table: Dim_ClassTerm

```
--Use the following truncate statement when reloading the table
--truncate table Dim_ClassTerm
```

```
insert into Dim_ClassTerm (
        [Institution_Key]
        ,Faculty_Service_Key
                ,Academic_Career_Key
                , Academic_Unit_Key
                , Subject_Key
                , Course_Key
                ,Academic_Term_Code
                ,Modality_Key
                ,Class_Term_Key
                ,Session_Code
                , Faculty_Service_Name_EN
                , Faculty_Service_Name_FR
                , Academic_Career_Name_EN
                , Academic_Career_Name_FR
                , Academic_Unit_Name_EN
                , Academic_Unit_Name_FR
                , Subject_Name_EN
                , Subject_Name_FR
                , Course_Code
                , Course_Name_EN
                , Course_Name_FR
                ,uoCampus_Class_Nbr
                ,Class_Start_Date
                ,Class_End_Date
                ,Class_SSR_Component
                ,Class_Cancel_Date
                ,Class_Section
                ,Class_Language
                ,Instruction_Mode
                ,Modality_Description
)
--Data Retrieval Statement
select  D.[Institution_Key]
                , C.Faculty_Service_Key
                , E.Academic_Career_Key
                , B.Academic_Unit_Key
                , A.Subject_Key
                , D.Course_Key
                ,F.Academic_Term_Code
```

```
43                  ,F.Modality_Key
44                  ,F.Class_Term_Key
45                  ,F.Session_Code
46                  , C.Faculty_Service_Name_EN
47                  , C.Faculty_Service_Name_FR
48                  , E.Academic_Career_Name_EN
49                  , E.Academic_Career_Name_FR
50                  , B.Academic_Unit_Name_EN
51                  , B.Academic_Unit_Name_FR
52                  , A.Subject_Name_EN
53                  , A.Subject_Name_FR
54                  , A.Subject_Code+ D.Catalog_Number as CourseCode
55                  , D.Course_Name_EN
56                  , D.Course_Name_FR
57                  ,F.uoCampus_Class_Nbr
58                  ,F.Class_Start_Date
59                  ,F.Class_End_Date
60                  ,F.Class_SSR_Component
61                  ,F.Class_Cancel_Date
62                  ,F.Class_Section
63                  ,F.Class_Language
64                  ,G.Instruction_Mode
65                  ,G.Modality_Description
66
67  from
68          (((((uocampus].[Subject_d] A
69          left join [uocampus].[Academic_Unit_d] B on (A.
               Academic_Unit_Key=B.Academic_Unit_Key and A.
               Institution_Key=B.Institution_Key))
70          left join [common].[Faculty_Service_d] C on (B.
               Faculty_Service_Key=C.Faculty_Service_Key and  B.
               Institution_Key=C.Institution_Key))
71          right join [uocampus].[Course_d] D on (A.Institution_Key=D
               .Institution_Key and A.Subject_Key=D.Subject_Key))
72          left join [uocampus].[Academic_Career_d] E on (D.
               Course_Academic_Career_Key=E.Academic_Career_Key))
73          right join [uocampus].[Class_Term_d] F on (D.Course_Key=F.
               Course_Key))
74          left join [uocampus].[Modality_d] G on (F.Modality_Key=G.
               Modality_Key)
```

```
75
76 order by C.[Institution_Key]
77                  , C.Faculty_Service_Key
78                  , E.Academic_Career_Key
79                  , B.Academic_Unit_Key
80                  , A.Subject_Key
81                  , D.Course_Key
82                  ,F.Academic_Term_Code
83                  ,F.Modality_Key
```

Listing B.9: Inserting and Retrieving Scripting for dimension table: Dim_ClassTerm

B.0.1.6 Dim_Student

```
1  CREATE TABLE dbo.[Dim_Student](
2          Dim_Student_Key int IDENTITY(1,1) NOT NULL,
3          [Student_Key] [int]  NOT NULL,
4          [Student_Date_of_Birth] [int] NULL,
5          [Student_Gender] [nvarchar](1) NULL,
6          [Student_Preferred_Language] [nvarchar](max) NULL,
7          [From_Country_Key] [int] NULL,
8          [Citizenship_Visa_Status_Key] [int] NULL,
9          [Student_Maternal_Language] [nvarchar](max) NULL,
10         [Student_Language_In_Use] [nvarchar](max) NULL,
11         Start_Residency_Status [nvarchar](max) NULL,
12         From_Country_Name_EN [nvarchar](max) NULL,
13         Citizenship_Status_Name_EN [nvarchar](max) NULL,
14         Citizenship_Status_Name_FR [nvarchar](max) NULL,
15         Visa_Status_Name_EN [nvarchar](max) NULL,
16         Visa_Status_Name_FR[nvarchar](max) NULL,
17         Program_Register_EffectiveDate int null,
18         Start_Term_Code int null,
19         FirstProgramRegister_Key int null,
20          FirstPlanRegister_Key int null
21  CONSTRAINT [PK_Dim_Student] PRIMARY KEY CLUSTERED
22 (
23         [Dim_Student_Key] ASC
24 )WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

```
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

Listing B.10: Creation Scripting for dimension table: Dim_Student

```
--Use the following truncate statement when reloading the table
--truncate table dbo.Dim_Student;
with tmp_firstTermRegister AS
(SELECT *, row_number() over (partition by Student_Key order by
   Academic_Term_Code) as TermNo
  FROM [uocampus].[Student_Program_Enrollment_f])
 ,

tmp_firstTerm AS(
 select *
 from tmp_firstTermRegister
where TermNo=1)


insert into dbo.Dim_Student([Student_Key],
        [Student_Date_of_Birth] ,
        [Student_Gender] ,
        [Student_Preferred_Language],


        [Student_Maternal_Language],
        [Student_Language_In_Use],
        Start_Residency_Status,
        [From_Country_Key],
        From_Country_Name_EN ,
        [Citizenship_Visa_Status_Key] ,
        Citizenship_Status_Name_EN ,
        Citizenship_Status_Name_FR ,
        Visa_Status_Name_EN ,
        Visa_Status_Name_FR,
        Program_Register_EffectiveDate,
        Start_Term_Code,
        FirstProgramRegister_Key,
         FirstPlanRegister_Key)
--Data Retrieval Statement
select A.[Student_Key],
```

```
36          A.[Student_Date_of_Birth] ,
37          A.[Student_Gender] ,
38          A.[Student_Preferred_Language],
39
40
41          A.[Student_Maternal_Language],
42          A.[Student_Language_In_Use],
43          B.Residency, B.Country_Key as From_Country_Key, C.
               Country_Name_EN as From_Country_Name_EN, D.
               Citizenship_Visa_Status_Key,D.
               Citizenship_Status_Name_EN ,
44          D.Citizenship_Status_Name_FR ,
45          D.Visa_Status_Name_EN ,
46          D.Visa_Status_Name_FR,
47          B.Effective_Date As Program_Register_EffectiveDate,
48          B.Academic_Term_Code As Start_Term_Code,
49          B.Academic_Program_Key as FirstProgramRegister_Key,
50          B.Academic_Plan_Key as FirstPlanRegister_Key
51  from [uocampus].[Student_d] A
52  left join tmp_firstTerm B on (A.Student_Key=B.Student_Key)
53  left join [common].[Country_d] C on (B.Country_Key=C.Country_Key)
54  left join [uocampus].[Citizenship_Visa_Status_d] D on (A.
       Citizenship_Visa_Status_Key=D.Citizenship_Visa_Status_Key)
```

Listing B.11: Inserting and Retrieving Scripting for dimension table: Dim_Student

B.0.2 DW Layer-Developing Fact Tables

B.0.2.1 Fact_Student_Class_Term

```
1   create TABLE dbo.[Fact_Student_Class_Term_Enrollment](
2           [Fact_Student_Term_Class_Enrollment_Key] [int] IDENTITY
               (1,1) NOT NULL,
3           [Student_Term_Class_Enrollment_Key] [int] NOT NULL,
4
5           [Class_Term_Units_Taken] [decimal](4, 2) NULL,
6           [Class_Term_Enrollment_Status] [nvarchar](50) NULL,
7           [Class_Term_Enrollment_DropDate] [int] NULL,
8           [Course_Grade_Official] [nvarchar](max) NULL,
```

```
        [Course_Grade_Input] [nvarchar](max) NULL,
        [Course_Grade_Date] [datetime] NULL,
        [Course_Grading_Basis] [nvarchar](max) NULL,
        [Course_Grading_Basis_Override] [nvarchar](max) NULL,
        [Course_Grade_Earn_Credit] [nvarchar](max) NULL,
        [Course_Grade_Include_In_GPA] [nvarchar](max) NULL,
        [Course_Grade_Units_Attempted] [nvarchar](max) NULL,
        [Course_Grade_Points] [decimal] NULL,
        [Course_Grade_Points_Per_Unit] [decimal] NULL,
        Dim_Student_Key [int] NULL ,
        [Dim_ClassEnrollment_Key] [int] NULL ,
        [Academic_Term_Code] [int] NULL,
        Dim_Program_Key [int] NULL,
        Academic_Plan_Key [int] NULL,
        Dim_ProgramPlan_Key [int] NULL,
        Dim_CourseInfo_Key [int] NULL,
         Student_Term_Number [int] NULL,
                Student_Term_Key[int] NULL
 CONSTRAINT [PK_Fact_Student_Term_Class_Enrollment_f] PRIMARY KEY
    CLUSTERED
(
        [Fact_Student_Term_Class_Enrollment_Key] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
   OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY];
```

Listing B.12: Creation Scripting for fact table: Fact_Student_Class_Term

```
--Use the following truncate table when reloading the table
--truncate table dbo.[Fact_Student_Class_Term_Enrollment];
with tmp_TermNumber as(
SELECT distinct Student_Key ,  Academic_Term_Code,
   Student_Term_Key
 FROM [uocampus].[Student_Term_f]),

 tmp_FactTermNumber as(

 select * ,DENSE_RANK() over (Partition by Student_Key order by
    Academic_Term_Code) as Student_TermNumber
 from  tmp_TermNumber ),

```

```
 tmp_StudentTermClass as(
 select A.*, B.Student_TermNumber
 from [uocampus].[Student_Class_Term_Enrollment_f] A left join
    tmp_FactTermNumber B on (A.Student_Key =B.Student_Key AND A.
    Student_Term_Key=B.Student_Term_Key))

insert into dbo.[Fact_Student_Class_Term_Enrollment](
        [Student_Term_Class_Enrollment_Key],
        [Class_Term_Units_Taken],
        [Class_Term_Enrollment_Status],
        [Class_Term_Enrollment_DropDate] ,
        [Course_Grade_Official],
        [Course_Grade_Input],
        [Course_Grade_Date],
        [Course_Grading_Basis] ,
        [Course_Grading_Basis_Override] ,
        [Course_Grade_Earn_Credit],
        [Course_Grade_Include_In_GPA] ,
        [Course_Grade_Units_Attempted] ,
        [Course_Grade_Points] ,
        [Course_Grade_Points_Per_Unit],
        Dim_Student_Key  ,
        [Dim_ClassEnrollment_Key],
        [Academic_Term_Code],
        Dim_Program_Key,
        Academic_Plan_Key,
        Dim_ProgramPlan_Key ,
        Dim_CourseInfo_Key,
        Student_Term_Number,
        Student_Term_Key)
--Data Retrieval Statement
select A.[Student_Term_Class_Enrollment_Key],
        A.[Class_Term_Units_Taken],
        A.[Class_Term_Enrollment_Status],
        A.[Class_Term_Enrollment_DropDate] ,
        A.[Course_Grade_Official],
        A.[Course_Grade_Input],
        A.[Course_Grade_Date],
        A.[Course_Grading_Basis] ,
        A.[Course_Grading_Basis_Override] ,
```

```
50          A.[Course_Grade_Earn_Credit],
51          A.[Course_Grade_Include_In_GPA] ,
52          A.[Course_Grade_Units_Attempted] ,
53          A.[Course_Grade_Points] ,
54          A.[Course_Grade_Points_Per_Unit],
55          B.Dim_Student_Key , C.[Dim_ClassEnrollment_Key] ,D.[
                Academic_Term_Code],E.Dim_Program_Key, F.
                Academic_Plan_Key,G.Dim_ProgramPlan_Key,H.
                Dim_CourseInfo_Key, A.Student_TermNumber, A.
                Student_Term_Key
56
57  from
58  /*Join Dim_Student on 1 Key Student_Key*/
59   tmp_StudentTermClass A left join [dbo].[Dim_Student] B on (A.
        Student_Key=B.Student_Key)
60  /*Join Dim_ClassTerm on ClassTermKey */
61  left join [dbo].[Dim_ClassTerm] C on (A.Class_Term_Key=C.
       Class_Term_Key)
62  /*Join [dbo].[Dim_AcademicTermCode on Academic_Term_Code */
63  left join [dbo].[Dim_AcademicTermCode] D on (C.[Academic_Term_Code
       ]=D.[Academic_Term_Code])
64  /*Join [dbo].[Dim_Program] on Academic_Program_Key */
65  left join [dbo].[Dim_Program] E on A.Academic_Program_Key=E.
       Academic_Program_Key
66  /*Join [uocampus].[Student_Program_Enrollment_f] on 3 Keys
       Academic_Program_Key, Academic_Plan_Key, Academic_Term_Code,
       Student_Key*/
67  left join [uocampus].[Student_Program_Enrollment_f] F on (E.
       Academic_Program_Key=F.Academic_Program_Key and A.Student_Key=F
       .Student_Key and D.[Academic_Term_Code]=F.Academic_Term_Code)
68  left join [dbo].[Dim_Plan] G on (F.Academic_Plan_Key=G.
       Academic_Plan_Key)
69  /*Join [dbo].[Dim_CourseInfo] by Course Key from Dim_ClassTerm on
       CourseKey*/
70  left join [dbo].[Dim_CourseInfo] H on (C.Course_Key=H.Course_Key)
```

Listing B.13: Inserting and Retrieving Scripting for Fact table: Fact_Student_Class_Term

B.0.2.2 Fact_StudentProgramEnrollment

```
1
2  CREATE TABLE [dbo].[Fact_StudentProgramEnrollment](
3          [Fact_StudentProgramEnrollment_Key] [int] IDENTITY(1,1)
             NOT NULL,
4          [Student_Program_Enrollment_Key] int NOT NULL,
5          Dim_Student_Key int Not null,
6          Dim_Program_Key int Not null,
7          [Student_Key] [int] NOT NULL,
8          [Academic_Term_Code] [int] NULL,
9          [Academic_Program_Key] [int] NULL,
10         [Academic_Plan_Key] [int] NULL,
11         [Program_Language] [nvarchar](max) NULL,
12         [Current_Year_of_Study] [nvarchar](max) NULL,
13         [Effective_Date] [int] NULL,
14         [Residency] [nvarchar](max) NULL,
15         [Country_Key] [int] NULL,
16         [Student_Term_Key] [int] NULL,
17         TermNumber int NOT NULL
18  CONSTRAINT [PK_Fact_StudentProgramEnrollment] PRIMARY KEY
      CLUSTERED
19 (
20         [Fact_StudentProgramEnrollment_Key] ASC
21 )WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
     OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
22 ) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
23 GO
```

Listing B.14: Creation Scripting for fact table: Fact_StudentProgramEnrollment

```
1  --Use the following truncate statement for reloading the data
     table
2  --Truncate table  [dbo].[Fact_StudentProgramEnrollment];
3  Insert into [dbo].[Fact_StudentProgramEnrollment] (
4                  Dim_Student_Key
5           ,Dim_Program_Key
6            ,[Student_Program_Enrollment_Key]
7        ,[Student_Key]
8        ,[Academic_Term_Code]
9        ,[Academic_Program_Key]
```

```
10        ,[Academic_Plan_Key]
11        ,[Program_Language]
12        ,[Current_Year_of_Study]
13        ,[Effective_Date]
14        ,[Residency]
15        ,[Country_Key]
16        ,[Student_Term_Key]
17            ,TermNumber
18  )
19  --Retrieval Statement
20  SELECT  B.Dim_Student_Key
21                  ,C.Dim_Program_Key
22            ,A.[Student_Program_Enrollment_Key]
23        ,A.[Student_Key]
24        ,A.[Academic_Term_Code]
25        ,A.[Academic_Program_Key]
26        ,A.[Academic_Plan_Key]
27        ,A.[Program_Language]
28        ,A.[Current_Year_of_Study]
29        ,A.[Effective_Date]
30        ,A.[Residency]
31        ,A.[Country_Key]
32        ,A.[Student_Term_Key]
33            ,ROW_NUMBER() over (partition by  A.[Student_Key]
34         order by A.[Academic_Term_Code] asc) as TermNumber
35    FROM (([uocampus].[Student_Program_Enrollment_f] A
36    left join [dbo].[Dim_Student] B on (A.Student_Key=B.Student_Key)
        ))
37    left join [dbo].[Dim_Program] C on (A.Academic_Program_Key=C.
        Academic_Program_Key)
38    order by A.[Academic_Term_Code]
```

Listing B.15: Inserting and Retieving Scripting for fact table: Fact_Student_StudentProgramEnrollment

B.0.3 Data Mart Layer for Learning Analytics Project- Developing Aggregated Tables

B.0.3.1 dbo.Aggregate_Student_Class_Enrollment

```
CREATE TABLE dbo.Aggregate_Student_Class_Enrollment (
      Student_Term_Class_Enrollment_Key int not null,
  Dim_Student_Key INT,
      StudentRegistered_Institution_Key INT,
      StudentRegistered_Registered_Faculty_Service_Key INT,
      StudentRegistered_Registered_Program_Key INT,
      StudentRegistered_Registered_Plan_Key INT,
      Program_Name_EN varchar(50),
      Plan_Language varchar(50),
      Academic_Career_Key INT,
  Class_StartTerm INT,
  Class_TermCode INT,
  Student_Term_Number INT,
  Student_Gender VARCHAR(10),
  Start_Residency_Status VARCHAR(50),
  From_Country_Name_EN VARCHAR(100),
  Student_StartTerm INT,
  Instruction_Mode VARCHAR(50),
  Modality_Description VARCHAR(100),
  Course_Name_EN VARCHAR(255),
  Course_Code VARCHAR(50),
  Subject_Name_EN VARCHAR(255),
  Subject_Key INT,
  Class_Language VARCHAR(50),
  Current_Age INT,
  Program_Credits_Required DECIMAL(5,2),
  Program_Normal_Completion INT,
  Diploma_Description_EN VARCHAR(255),
  Study_Field VARCHAR(255),
  Degree_Name_Formal_EN VARCHAR(255),
  Years_of_Education VARCHAR(50),
  Form_of_Study VARCHAR(100),
  Academic_Load VARCHAR(50),
  Total_Units_Term DECIMAL(5,2),
  Total_Units_Cumulative DECIMAL(8,2),
```

```
    Current_Year_of_Study INT,
    Current_Residency VARCHAR(100),
    CourseName VARCHAR(255),
    GradeGroup VARCHAR(50),
    Course_NoOfAttempt INT,
    Cumm_PreviousTermEarnedUnitRates DECIMAL(5,2),
    Cumm_PreviousTermPercentComplete DECIMAL(5,2),
    Cumm_PreviousTermGPA DECIMAL(4,2),
    Cumm_PreviousTermPassedRates DECIMAL(5,2),
    Cumm_PreviousTermAtRiskRates DECIMAL(5,2),
    Cumm_PreviousTermDroppedRates DECIMAL(5,2),
    Course_Grading_Basis VARCHAR(50),
    Course_Grading_Basis_Override VARCHAR(50),
    Course_Grade_Earn_Credit CHAR(1),
    Course_Grade_Include_In_GPA CHAR(1),
    Course_Grade_Units_Attempted CHAR(1),
    Course_Grade_Points DECIMAL(5,2),
    Course_Grade_Points_Per_Unit DECIMAL(5,2),
    Course_Grade_Official VARCHAR(50),
    Class_Term_Enrollment_DropDate INT,
    Class_Term_Enrollment_Status VARCHAR(50),
    Class_Term_Units_Taken DECIMAL(5,2),
    AVG_PreviousTermNoOfRegisteredUnits float
);
```

Listing B.16: Creation Scripting for aggregation table: dbo.Aggregate_Student_Class_Enrollment

```
--Use the following truncate table when reloading the table
--truncate table [dbo].[Aggregate_Student_Class_Enrollment]
with tmp as(
SELECT
A.[Student_Term_Class_Enrollment_Key],
dbo.Dim_ClassTerm.Class_Term_Key,
A.Dim_Student_Key,
dbo.Dim_Program.[Institution_Key] as
   StudentRegistered_Institution_Key,
dbo.Dim_Program.[Faculty_Service_Key] as
   StudentRegistered_Registered_Faculty_Service_Key,
dbo.Dim_Program.[Dim_Program_Key] as
   StudentRegistered_Registered_Program_Key,
```

```
11 dbo.Dim_Student.FirstPlanRegister_Key as
      StudentRegistered_Registered_Plan_Key,
12 dbo.Dim_Program.Program_Name_EN as
      StudentRegistered_Program_Name_EN,
13 uocampus.Academic_Plan_d.[Academic_Plan_Language] as
      StudentRegistered_Plan_Language,
14 dbo.Dim_Program.Academic_Career_Key as
      StudentRegistered_Academic_Career_Key ,
15 A.Course_Grading_Basis,
16 A.Course_Grading_Basis_Override,
17 A.Course_Grade_Earn_Credit,
18 A.Course_Grade_Include_In_GPA,
19 A.Course_Grade_Units_Attempted,
20 A.Course_Grade_Points,
21 A.Course_Grade_Points_Per_Unit,
22 A.Course_Grade_Official,
23 A.Class_Term_Enrollment_DropDate,
24 A.Class_Term_Enrollment_Status,
25 A.Class_Term_Units_Taken,
26 Dim_ClassTermCode.Academic_Term_Year_EN As Class_StartTerm,
27 Dim_ClassTermCode.Academic_Term_Code As Class_TermCode,
28 A.Student_Term_Number,
29 dbo.Dim_Student.Student_Gender,
30 dbo.Dim_Student.Start_Residency_Status,
31 dbo.Dim_Student.From_Country_Name_EN,
32 dbo.Dim_Student.Start_Term_Code As Student_StartTerm,
33 dbo.Dim_ClassTerm.Instruction_Mode,
34 dbo.Dim_ClassTerm.Modality_Description,
35 dbo.Dim_ClassTerm.Course_Name_EN,
36 dbo.Dim_ClassTerm.Course_Code,
37 dbo.Dim_ClassTerm.Subject_Name_EN,
38 dbo.Dim_ClassTerm.Subject_Key,
39 Dim_ClassTerm.Class_Language,
40 Datediff(year,convert(nvarchar(10),dbo.Dim_Student.
      Student_Date_of_Birth,120),convert(nvarchar(10),Dim_ClassTerm.[
      Class_Start_Date],120)) as Current_Age,
41 dbo.Dim_Program.Program_Credits_Required,
42 dbo.Dim_Program.Program_Normal_Completion,
43 uocampus.Academic_Plan_d.Diploma_Description_EN,
44 uocampus.Academic_Plan_d.Study_Field,
```

```
45 uocampus.Degree_d.Degree_Name_Formal_EN,
46 uocampus.Degree_d.Years_of_Education,
47 Student_Term.[Form_of_Study],
48 Student_Term.[Academic_Load],
49 Student_Term.[Total_Units_Term],
50 Student_Term.[Total_Units_Cumulative],
51 Student_Term.[Current_Year_of_Study],
52 Student_Term.[Residency] as Current_Residency,
53 Dim_ClassTerm.Course_Code + ' ' + Dim_ClassTerm.Course_Name_EN As
      CourseName,
54 --Gradegroup for undergrad, engineering
55     CASE
56         WHEN A.Class_Term_Enrollment_Status = 'W' THEN 'No Attempt
              '
57
58         -- Empty value grades
59         WHEN A.Course_Grade_Official = '' THEN
60             CASE
61                 WHEN A.Class_Term_Enrollment_Status = 'E' AND A.
                      Course_Grade_Units_Attempted = 'N' THEN 'No
                      Attempt'
62                 WHEN A.Class_Term_Enrollment_Status = 'D' AND A.
                      Course_Grade_Units_Attempted IN ('Y', 'I') THEN
                       'Drop After Attempt'
63                 WHEN A.Class_Term_Enrollment_Status = 'D' AND A.
                      Course_Grade_Units_Attempted = 'N' THEN 'Drop
                      Before Attempt'
64                 WHEN A.Class_Term_Enrollment_Status = 'E' AND A.
                      Course_Grade_Units_Attempted = 'I' THEN 'Not
                      Updated'
65                 ELSE 'Unknown'
66             END
67
68         -- Not Empty Grade values
69         ELSE
70             CASE
71                 WHEN A.Class_Term_Enrollment_Status = 'D' AND A.
                      Course_Grade_Units_Attempted IN ('Y', 'I') THEN
                       'Drop After Attempt'
72                 WHEN A.Class_Term_Enrollment_Status = 'D' AND A.
```

```
                    Course_Grade_Units_Attempted = 'N' THEN 'Drop
                    Before Attempt'
73                WHEN A.Course_Grade_Official IN ('A+', 'A') THEN '
                    Excellent'
74                WHEN A.Course_Grade_Official IN ('A-', 'B+', 'B',
                    'P', 'S','C','C+') THEN 'Passed'
75                                WHEN A.Course_Grade_Official IN (
                                    'D+', 'D') THEN 'At-risk'
76                WHEN A.Course_Grade_Official IN ( 'E', 'F', 'EIN',
                     'NS','ABS', 'INC') THEN 'Failed'
77                WHEN A.Course_Grade_Official IN ( 'DR') THEN 'Drop
                     After Attempt'
78                WHEN A.Course_Grade_Official IN ('CR', 'Q', 'AUD',
                     'NC') THEN 'Audit/No Units/OtherInstitution'
79                WHEN A.Course_Grade_Official IN ('DNW', 'CTN', '
                    DFR', 'AEC') THEN 'Continue/Deferred'
80                WHEN A.Course_Grade_Official = 'NNR' THEN 'Not
                    Updated'
81                ELSE 'Unknown Grades'
82            END
83    END AS GradeGroup,
84
85        DENSE_RANK() OVER (
86        PARTITION BY A.Dim_Student_Key, A.Dim_CourseInfo_Key
87        ORDER BY A.Student_Term_Number ASC
88    ) AS Course_NoOfAttempt
89
90 --SELECT COUNT(*)
91 FROM          ( uocampus.Degree_d INNER JOIN
92              uocampus.Academic_Plan_d ON uocampus.Degree_d.
                    Degree_Key = uocampus.Academic_Plan_d.Degree_Key)
                    RIGHT JOIN
93              (dbo.Dim_ClassTerm left JOIN dbo.
                    Fact_Student_Class_Term_Enrollmentas A ON dbo.
                    Dim_ClassTerm.Dim_ClassEnrollment_Key = A.
                    Dim_ClassEnrollment_Key LEFT JOIN
94              dbo.Dim_Student ON A.Dim_Student_Key = dbo.Dim_Student
                    .Dim_Student_Key LEFT JOIN
95              dbo.Dim_Program ON dbo.Dim_Student.
                    FirstProgramRegister_Key = dbo.Dim_Program.
```

```
                Academic_Program_Key )
                        ON
    uocampus.Academic_Plan_d.Academic_Plan_Key = dbo.Dim_Student.
       FirstPlanRegister_Key LEFT OUTER JOIN
    dbo.Dim_AcademicTermCode ON dbo.Dim_Student.Start_Term_Code =
       dbo.Dim_AcademicTermCode.Academic_Term_Code LEFT JOIN
    dbo.Dim_CourseInfo ON A.Dim_CourseInfo_Key = dbo.
       Dim_CourseInfo.Dim_CourseInfo_Key LEFT JOIN
    dbo.Dim_AcademicTermCode AS Dim_ClassTermCode ON A.
       Academic_Term_Code = Dim_ClassTermCode.Academic_Term_Code
        LEFT JOIN
        [uocampus].[Student_Term_f] as Student_Term ON A.
           Student_Term_Key = Student_Term.Student_Term_Key
WHERE
--Instituion = 'University of Ottawa'
(dbo.Dim_Program.Institution_Key = 1)
AND (dbo.Dim_Program.Faculty_Service_Key = 7)
AND (dbo.Dim_Program.Academic_Career_Key = 8)
----Program: Faculty Engineering, Undergrad, English
--AND   (uocampus.Academic_Plan_d.Academic_Plan_Language = 'EN')
--AND (dbo.Dim_Program.Program_Name_EN = 'BASc-Eng')
----Remove  Enroll='W'
AND (A.Class_Term_Enrollment_Status <> 'W')
---- Not drop before attempt or not join class or audit only
),
---Check valid courses
dataset as(
select *
from tmp
where GradeGroup NOT IN ('No Attempt', 'Drop Before Attempt', 'Not
    Updated', 'Audit/No Units/OtherInstitution', 'Unknown Grades')
    )
--where --All class start since Fall 2021 to Summer 2023
--( Class_TermCode >= '2219') AND ( Class_TermCode < '2239')
--Calculating measurements
--where --All class start since Fall 2021 to Summer 2023
--( Class_TermCode >= '2219') AND ( Class_TermCode < '2239')
--CREATE CALCULATED MEASURES
--Previous term cumulative GPA of all courses whose GPA included

```

```
128 --Previous term cumulative Pass rated
129
130 --Previous term cumulative Dropped Rated
131 --Previous term cumulative Earned Credits
132
133 --Previous_Cumulative_Credit as(
134 ,
135 Student_sumtable as(
136 select
137     --CASE
138     --    WHEN GradeGroup IN ('No Attempt', 'Drop Before Attempt',
        'Not Updated',
139     --                        'Unknown', 'Audit/No Units/
        OtherInstitution',
140     --                        'No Units', 'Unknown Grades')
141     --    THEN 0
142     --    ELSE 1
143     --END AS ValidCourse,
144         Dim_Student_Key, Student_Term_Number,
145         SUM( case when [Course_Grade_Earn_Credit]='Y' and
            Course_Grade_Units_Attempted='Y' then [
            Class_Term_Units_Taken] else 0 end)   AS
            TermEarnedUnits,
146         SUM(case when [Course_Grade_Include_In_GPA]='Y' and
            Course_Grade_Units_Attempted='Y' then [
            Class_Term_Units_Taken]*[Course_Grade_Points_Per_Unit]
            else 0 end) as TermEarnedPoints,
147         SUM(case when [Course_Grade_Include_In_GPA]='Y' and
            Course_Grade_Units_Attempted='Y' then [
            Class_Term_Units_Taken] else 0 end) as TermTotalUnits,
148
149         --Number of registed courses which is valid
150         SUM(case WHEN GradeGroup IN ('No Attempt', 'Drop Before
            Attempt', 'Not Updated',
151                         'Unknown', 'Audit/No Units/
                              OtherInstitution',
152                         'No Units', 'Unknown Grades')
153         THEN 0
154         ELSE 1 end) as NoOfCourses,
155 --No Of Passed courses
```

```
156         SUM(case when GradeGroup='Passed' or  GradeGroup='
               Excellent' then 1 else 0 end) as NoOfPassedCourses,
157 --No Of Dropped courses
158         SUM(case when GradeGroup='Drop After Attempt'  then 1 else
                0 end) as NoOfDroppedCourses ,
159 --No Of Failed courses
160         SUM(case when GradeGroup='Failed' or GradeGroup='At-risk'
                then 1 else 0 end) as NoOfAtRiskCourses
161
162 from tmp
163 where ([Course_Grade_Earn_Credit]='Y' or [
       Course_Grade_Include_In_GPA]='Y')  and
       Course_Grade_Units_Attempted='Y'
164 group by Dim_Student_Key, Student_Term_Number),
165 --select * from tmp where  Dim_Student_Key='8757' and
       Course_Grade_Units_Attempted='Y'
166 PreviousTerm_Cumulative aS(
167 select Dim_Student_Key, Student_Term_Number,
168         sum(TermEarnedUnits)
169         over (partition by Dim_Student_Key order by
               Student_Term_Number   ROWS BETWEEN UNBOUNDED PRECEDING
               AND 1 PRECEDING ) as CumPreviousTerm_EarnedUnits,
170         sum(TermEarnedPoints)
171         over (partition by Dim_Student_Key order by
               Student_Term_Number   ROWS BETWEEN UNBOUNDED PRECEDING
               AND 1 PRECEDING )
172         as CumPreviousTerm_EarnedPoints,
173         sum(TermTotalUnits)
174         over (partition by Dim_Student_Key order by
               Student_Term_Number   ROWS BETWEEN UNBOUNDED PRECEDING
               AND 1 PRECEDING )
175         as CumPreviousTerm_TotalUnits,
176         sum(NoOfPassedCourses)
177         over (partition by Dim_Student_Key order by
               Student_Term_Number   ROWS BETWEEN UNBOUNDED PRECEDING
               AND 1 PRECEDING )
178         as CumPreviousTerm_NoOfPassedCourses,
179         sum(NoOfDroppedCourses)
180         over (partition by Dim_Student_Key order by
               Student_Term_Number   ROWS BETWEEN UNBOUNDED PRECEDING
```

```
            AND 1 PRECEDING )
        as CumPreviousTerm_NoOfDroppedCourses ,
        sum(NoOfAtRiskCourses)
        over (partition by Dim_Student_Key order by
            Student_Term_Number   ROWS BETWEEN UNBOUNDED PRECEDING
            AND 1 PRECEDING )
        as CumPreviousTerm_NoOfAtRiskCourses ,
        sum(NoOfCourses)
        over (partition by Dim_Student_Key order by
            Student_Term_Number   ROWS BETWEEN UNBOUNDED PRECEDING
            AND 1 PRECEDING )
        as CumPreviousTerm_NoOfCourses



from student_sumtable)
,

course_sumtable as(
select
        Course_Code,Class_TermCode,
        sum(case when GradeGroup='Failed' then 1 else 0 end) as
            NoOfFailed,
        count(Dim_Student_Key) as NoOfRegistered,
        sum(case when GradeGroup='Failed' then 1 else 0 end)*1.0/
            count(Dim_Student_Key) as FailedRates

from tmp
where GradeGroup NOT IN ('No Attempt', 'Drop Before Attempt', 'Not
    Updated',
                            'Unknown', 'Audit/No Units/
                               OtherInstitution',
                            'No Units', 'Unknown Grades')
group by Course_Code,Class_TermCode
)
,
PreviousTerm_CourseAvg as(
select *, AVG(FailedRates) over (partition by Course_Code order by
    Class_TermCode asc ROWS BETWEEN UNBOUNDED PRECEDING AND 1
   PRECEDING)  as AVG_TillPreviousTermFailedRates
```

```
210 from course_sumtable)
211 ,
212 agg_dataset as(
213 select curr.*,
214                 prev.CumPreviousTerm_EarnedUnits*1.0/nullif(prev.
                       CumPreviousTerm_TotalUnits,0) as
                       Cumm_PreviousTermEarnedUnitRates,
215                 prev.CumPreviousTerm_EarnedUnits*1.0/nullif(curr.
                       Program_Credits_Required,0) as
                       Cumm_PreviousTermPercentComplete,
216                 prev.CumPreviousTerm_EarnedPoints*1.0/nullif(prev.
                       CumPreviousTerm_TotalUnits,0) as
                       Cumm_PreviousTermGPA,
217                 prev.CumPreviousTerm_NoOfPassedCourses*1.0/nullif(
                       prev.CumPreviousTerm_NoOfCourses,0) as
                       Cumm_PreviousTermPassedRates,
218                 prev.CumPreviousTerm_NoOfAtRiskCourses*1.0 /
                       nullif(prev.CumPreviousTerm_NoOfCourses,0) as
                       Cumm_PreviousTermAtRiskRates,
219                 prev.CumPreviousTerm_NoOfDroppedCourses*1.0 /
                       nullif(prev.CumPreviousTerm_NoOfCourses,0) as
                       Cumm_PreviousTermDroppedRates,
220                 prev_course.AVG_TillPreviousTermFailedRates as
                       Course_PreviousTerm_AVGDroppedRates
221
222 from dataset curr
223         left join
224         PreviousTerm_Cumulative prev on curr.Dim_Student_Key=Prev.
               Dim_Student_Key and curr.Student_Term_Number=prev.
               Student_Term_Number
225         left join
226         PreviousTerm_CourseAvg prev_course on curr.Course_Code=
               prev_course.Course_Code and curr.Class_TermCode=
               prev_course.Class_TermCode
227 )
228 --Insert Statements
229 INSERT INTO dbo.Aggregate_Student_Class_Enrollment (
230     Student_Term_Class_Enrollment_Key,
231     Class_Term_Key,
232     Dim_Student_Key,
```

```
    StudentRegistered_Institution_Key,
    StudentRegistered_Registered_Faculty_Service_Key,
    StudentRegistered_Registered_Program_Key,
    StudentRegistered_Registered_Plan_Key,
    StudentRegistered_Program_Name_EN,
    StudentRegistered_Plan_Language,
    StudentRegistered_Academic_Career_Key,
    Class_StartTerm,
    Class_TermCode,
    Student_Term_Number,
    Student_Gender,
    Start_Residency_Status,
    From_Country_Name_EN,
    Student_StartTerm,
    Instruction_Mode,
    Modality_Description,
    Course_Name_EN,
    Course_Code,
    Subject_Name_EN,
    Subject_Key,
    Class_Language,
    Current_Age,
    Program_Credits_Required,
    Program_Normal_Completion,
    Diploma_Description_EN,
    Study_Field,
    Degree_Name_Formal_EN,
    Years_of_Education,
    Form_of_Study,
    Academic_Load,
    Total_Units_Term,
    Total_Units_Cumulative,
    Current_Year_of_Study,
    Current_Residency,
    CourseName,
    GradeGroup,
    Course_NoOfAttempt,
    Cumm_PreviousTermEarnedUnitRates,
    Cumm_PreviousTermPercentComplete,
    Cumm_PreviousTermGPA,
```

```
    Cumm_PreviousTermPassedRates,
    Cumm_PreviousTermAtRiskRates,
    Cumm_PreviousTermDroppedRates,
    Course_PreviousTerm_AVGDroppedRates,
    Course_Grading_Basis,
    Course_Grading_Basis_Override,
    Course_Grade_Earn_Credit,
    Course_Grade_Include_In_GPA,
    Course_Grade_Units_Attempted,
    Course_Grade_Points,
    Course_Grade_Points_Per_Unit,
    Course_Grade_Official,
    Class_Term_Enrollment_DropDate,
    Class_Term_Enrollment_Status,
    Class_Term_Units_Taken
)
--Data Retrieval Statement
select
 A.Student_Term_Class_Enrollment_Key,
    A.Class_Term_Key,
    A.Dim_Student_Key,
        A.StudentRegistered_Institution_Key,
    A.StudentRegistered_Registered_Faculty_Service_Key,
    A.StudentRegistered_Registered_Program_Key,
    A.StudentRegistered_Registered_Plan_Key,
    A.StudentRegistered_Program_Name_EN,
    A.StudentRegistered_Plan_Language,
    A.StudentRegistered_Academic_Career_Key,
    A.Class_StartTerm,
    A.Class_TermCode,
    A.Student_Term_Number,
    A.Student_Gender,
    A.Start_Residency_Status,
    A.From_Country_Name_EN,
    A.Student_StartTerm,
    A.Instruction_Mode,
    A.Modality_Description,
    A.Course_Name_EN,
    A.Course_Code,
    A.Subject_Name_EN,
```

```
313     A.Subject_Key,
314     A.Class_Language,
315     A.Current_Age,
316     A.Program_Credits_Required,
317     A.Program_Normal_Completion,
318     A.Diploma_Description_EN,
319     A.Study_Field,
320     A.Degree_Name_Formal_EN,
321     A.Years_of_Education,
322     A.Form_of_Study,
323     A.Academic_Load,
324     A.Total_Units_Term,
325     A.Total_Units_Cumulative,
326     A.Current_Year_of_Study,
327     A.Current_Residency,
328     A.CourseName,
329     A.GradeGroup,
330     A.Course_NoOfAttempt,
331     A.Cumm_PreviousTermEarnedUnitRates,
332     A.Cumm_PreviousTermPercentComplete,
333     A.Cumm_PreviousTermGPA,
334     A.Cumm_PreviousTermPassedRates,
335     A.Cumm_PreviousTermAtRiskRates,
336     A.Cumm_PreviousTermDroppedRates,
337     A.Course_PreviousTerm_AVGDroppedRates,
338     A.Course_Grading_Basis,
339     A.Course_Grading_Basis_Override,
340     A.Course_Grade_Earn_Credit,
341     A.Course_Grade_Include_In_GPA,
342     A.Course_Grade_Units_Attempted,
343     A.Course_Grade_Points,
344     A.Course_Grade_Points_Per_Unit,
345     A.Course_Grade_Official,
346     A.Class_Term_Enrollment_DropDate,
347     A.Class_Term_Enrollment_Status,
348     A.Class_Term_Units_Taken
349 from agg_dataset  A
```

Listing B.17: Inserting and Retrieving Scripting for Aggregation table: dbo.Aggregate_Student_Class_Enrollment

B.0.3.2 dbo.Aggregate_Brightspace_AllGrades

```
1
2   drop table if exists dbo.Aggregate_Brightspace_AllGrades;
3 CREATE TABLE dbo.Aggregate_Brightspace_AllGrades (
4     Student_Term_Class_Enrollment_Key INT PRIMARY KEY,
5     GradingSystem VARCHAR(50),
6     Class_Term_Key int, Student_Key int, Class_NoOfStudent int,
7     Avg_Grade_Phase1 DECIMAL(5, 2),
8     Avg_Avg_GradeGroup_Phase1 DECIMAL(5, 2),
9     Avg_Grade_Phase2 DECIMAL(5, 2),
10    Avg_Avg_GradeGroup_Phase2 DECIMAL(5, 2),
11    Avg_Grade_Phase3 DECIMAL(5, 2),
12    Avg_Avg_GradeGroup_Phase3 DECIMAL(5, 2)
13 );
```

Listing B.18: Creation Scripting for aggregation table: dbo.Aggregate_Brightspace_AllGrades

```
1  --Use the following truncate table when reloading the table
2  --truncate table dbo.Aggregate_Brightspace_AllGrades;
3  --This table will includes all grades data from Brightspace
       filtered only classes that have all 3 phases data
4  with tmp as(
5  SELECT A.*,
6  DATEDIFF(day,convert(nvarchar(8), B.Class_Start_Date, 112),convert
       (nvarchar(8), B.Class_End_Date, 112)) as Class_Duration,
7  case when
8      DATEDIFF(day,convert(nvarchar(8), B.Class_Start_Date, 112),A.
           Grade_Last_Modified) <= DATEDIFF(day,convert(nvarchar(8), B
           .Class_Start_Date, 112),convert(nvarchar(8), B.
           Class_End_Date, 112))*0.3 then 'Phase1'
9      when DATEDIFF(day,convert(nvarchar(8), B.Class_Start_Date,
           112),A.Grade_Last_Modified) <=DATEDIFF(day,convert(nvarchar
           (8), B.Class_Start_Date, 112),convert(nvarchar(8), B.
           Class_End_Date, 112))*0.5 then 'Phase2'
10      when DATEDIFF(day,convert(nvarchar(8), B.Class_Start_Date,
            112),A.Grade_Last_Modified) <=DATEDIFF(day,convert(
           nvarchar(8), B.Class_Start_Date, 112),convert(nvarchar
           (8), B.Class_End_Date, 112))*0.75 then 'Phase3'
11     else 'Final'
```

```
12 end as Phase
13 FROM ([uocampus].[Brightspace_AllGrades] A
14 left join
15   [uocampus].[Class_Term_d] B on A.Class_Term_Key = B.
       Class_Term_Key)
16   left join [dbo].[Fact_Student_Class_Term_Enrollment]
17     on A.Student_Term_Class_Enrollment_Key = [dbo].[
         Fact_Student_Class_Term_Enrollment].
         Student_Term_Class_Enrollment_Key
18 where
19   --Condition data for Brightspace Table
20   A.Student_Term_Class_Enrollment_Key is not NULL
21   --All record which Excluded_From_Final_Grade <> 'True'
22   --AND (A.Excluded_From_Final_Grade != 'True' or A.
       Excluded_From_Final_Grade is NULL)
23   AND A.Is_Active = 'True'
24   --Remove all records with No Grade_Value
25     and A.Grade_Value is not NULL
26     AND A.Points_Denominator is not NULL
27
28 --Remove all records whose Class has no unit (don't count in GPA
     and no need to monitor at-risk)
29     and ([dbo].[Fact_Student_Class_Term_Enrollment].
         Class_Term_Units_Taken <> 0 or [dbo].[
         Fact_Student_Class_Term_Enrollment].Class_Term_Units_Taken
         is not null)
30 --Remove Final calculated r ws which was auto created by
     Brightspace
31     and LOWER(A.Grade_Item_Name) not in
32     ('note finale calcul e', 'final calculated grade'
33         ,'note finale ajust e', 'final adjusted grade')
34 ),
35 tbl_ClassWithAllPhases as(
36     select distinct Class_Term_Key, Course_Key,
37     count(distinct [Student_Term_Class_Enrollment_Key]) as
         Class_NoOfStudent,
38     sum(case when Grade_Item_Weight is not null and cast (
         Grade_Item_Weight as float) <> 0.0 then 1 else 0 end) as
         NoOfWeightItem
39     from tmp
```

```
40      group by Class_Term_Key, Course_Key
41      having count(case when Phase = 'Phase1' then 1 end) > 0
42          and count(case when Phase = 'Phase2' then 1 end) > 0
43          and count(case when Phase = 'Phase3' then 1 end) >0),
44
45
46  tmp_final as(
47      select tmp.*,
48              b.Class_NoOfStudent,
49              case when b.NoOfWeightItem > 0 then 'Weight-based'
                    else 'Point-based' end as GradingSystem
50      from tmp
51      right join tbl_ClassWithAllPhases b on tmp.Class_Term_Key = b.
           Class_Term_Key
52
53  ),
54
55  --Weight-based system : use Weight
56  --Point-based system: Use Denominator
57  tmp_Phase1 as(
58      --There is at lease one grade item with weight is not null:
           replace null weight with 0
59      select distinct Student_Term_Class_Enrollment_Key,
           Class_Term_Key, Student_Key, Class_NoOfStudent,
           GradingSystem,
60          case when GradingSystem='Weight-based' then sum(cast(
               ISNULL(Grade_Item_Weight,0) as float)*cast(ISNULL(
               Points_Numerator,0) as float)
61          /nullif(cast(Points_Denominator as float),0))
62          else sum(cast(ISNULL(Points_Numerator,0) as float)) end
63          as TotalPoints,
64          case when GradingSystem='Weight-based' then Sum(cast(
               ISNULL(Grade_Item_Weight,0) as float))
65          else sum(cast(Points_Denominator as float)) end
66          as TotalWeight,
67          count([Grade_Item_Category_ID]) as NoOfCategory
68      from tmp_final
69      where
70      Phase in ('Phase1')
71      group by Student_Term_Class_Enrollment_Key, Class_Term_Key,
```

```
        Student_Key, Class_NoOfStudent, [Grade_Item_Category_ID],
        GradingSystem
72      )
73  ,
74  tmp_Phase2 as(
75      --There is at lease one grade item with weight is not null:
        replace null weight with 0
76      select distinct Student_Term_Class_Enrollment_Key,
        Class_Term_Key, Student_Key, Class_NoOfStudent, [
        Grade_Item_Category_ID],GradingSystem,
77          case when GradingSystem='Weight-based' then sum(cast(
            ISNULL(Grade_Item_Weight,0) as float)*cast(ISNULL(
            Points_Numerator,0) as float)
78          /nullif(cast(Points_Denominator as float),0))
79          else sum(cast(ISNULL(Points_Numerator,0) as float)) end
80          as TotalPoints,
81          case when GradingSystem='Weight-based' then Sum(cast(
            ISNULL(Grade_Item_Weight,0) as float))
82          else sum(cast(Points_Denominator as float)) end
83          as TotalWeight,
84          count([Grade_Item_Category_ID]) as NoOfCategory
85      from tmp_final
86      where
87      Phase in ('Phase1','Phase2')
88      group by Student_Term_Class_Enrollment_Key, Class_Term_Key,
        Student_Key, Class_NoOfStudent, [Grade_Item_Category_ID],
        GradingSystem
89      ),
90  tmp_Phase3 as(
91      --There is at lease one grade item with weight is not null:
        replace null weight with 0
92      select distinct Student_Term_Class_Enrollment_Key,
        Class_Term_Key, Student_Key, Class_NoOfStudent, [
        Grade_Item_Category_ID],GradingSystem,
93          case when GradingSystem='Weight-based' then sum(cast(
            ISNULL(Grade_Item_Weight,0) as float)*cast(ISNULL(
            Points_Numerator,0) as float)
94          /nullif(cast(Points_Denominator as float),0))
95          else sum(cast(ISNULL(Points_Numerator,0) as float)) end
96          as TotalPoints,
```

```
97          case when GradingSystem='Weight-based' then Sum(cast(
               ISNULL(Grade_Item_Weight,0) as float))
98          else sum(cast(Points_Denominator as float)) end
99          as TotalWeight,
100         count([Grade_Item_Category_ID]) as NoOfCategory
101
102
103     from tmp_final
104     where
105     Phase in ('Phase1','Phase2','Phase3')
106     group by Student_Term_Class_Enrollment_Key, Class_Term_Key,
           Student_Key, Class_NoOfStudent, [Grade_Item_Category_ID],
           GradingSystem
107
108
109     ),
110 tbl_avg_grade_Phase1 as(
111     select Student_Term_Class_Enrollment_Key, Class_Term_Key,
           Student_Key, Class_NoOfStudent, GradingSystem,
112     CASE
113     WHEN SUM(TotalWeight) = 0 THEN 0
114     ELSE sum(TotalPoints)/nullif(sum(TotalWeight),0)
115     END as Avg_Grade,
116     case when SUM(TotalWeight) = 0 then 0
117     else sum(TotalPoints)/nullif(sum(TotalWeight),0) end as
           Avg_Avg_GradeGroup
118     from tmp_Phase1
119     group by Student_Term_Class_Enrollment_Key, Class_Term_Key,
           Student_Key, Class_NoOfStudent, GradingSystem
120 )
121 ,tbl_avg_grade_Phase2 as(
122    select Student_Term_Class_Enrollment_Key, Class_Term_Key,
          Student_Key, Class_NoOfStudent, GradingSystem,
123     CASE
124     WHEN SUM(TotalWeight) = 0 THEN 0
125     ELSE sum(TotalPoints)/nullif(sum(TotalWeight),0)
126     END as Avg_Grade,
127     case when SUM(TotalWeight) = 0 then 0
128     else sum(TotalPoints)/nullif(sum(TotalWeight),0) end as
           Avg_Avg_GradeGroup
```

```
129     from tmp_Phase2
130     group by Student_Term_Class_Enrollment_Key, Class_Term_Key,
            Student_Key, Class_NoOfStudent, GradingSystem
131 )
132 ,tbl_avg_grade_Phase3 as(
133     select Student_Term_Class_Enrollment_Key, Class_Term_Key,
            Student_Key, Class_NoOfStudent, GradingSystem,
134     CASE
135     WHEN SUM(TotalWeight) = 0 THEN 0
136     ELSE sum(TotalPoints)/nullif(sum(TotalWeight),0)
137     END as Avg_Grade,
138     case when SUM(TotalWeight) = 0 then 0
139     else sum(TotalPoints)/nullif(sum(TotalWeight),0) end as
            Avg_Avg_GradeGroup
140
141     from tmp_Phase3
142     group by Student_Term_Class_Enrollment_Key, Class_Term_Key,
            Student_Key, Class_NoOfStudent, GradingSystem
143 )
144 ,tbl_avg_grade_All as(
145     select
146     a.Student_Term_Class_Enrollment_Key, a.Class_Term_Key, a.
            Student_Key, a.Class_NoOfStudent, a.GradingSystem,
147     c.Avg_Grade as Avg_Grade_Phase1, c.Avg_Avg_GradeGroup as
            Avg_Avg_GradeGroup_Phase1,
148     b.Avg_Grade as Avg_Grade_Phase2, b.Avg_Avg_GradeGroup as
            Avg_Avg_GradeGroup_Phase2,
149     a.Avg_Grade as Avg_Grade_Phase3, a.Avg_Avg_GradeGroup as
            Avg_Avg_GradeGroup_Phase3
150     from tbl_avg_grade_Phase3 a left join
151     tbl_avg_grade_Phase2 b on a.Student_Term_Class_Enrollment_Key
            = b.Student_Term_Class_Enrollment_Key
152     left join tbl_avg_grade_Phase1 c
153     on COALESCE(a.Student_Term_Class_Enrollment_Key,b.
            Student_Term_Class_Enrollment_Key) = c.
            Student_Term_Class_Enrollment_Key
154 )
155 --Insert Statements
156 INSERT INTO dbo.Aggregate_Brightspace_AllGrades (
157     Student_Term_Class_Enrollment_Key, Class_Term_Key, Student_Key
```

```
        , Class_NoOfStudent , GradingSystem ,
158     Avg_Grade_Phase1 ,
159     Avg_Avg_GradeGroup_Phase1 ,
160     Avg_Grade_Phase2 ,
161     Avg_Avg_GradeGroup_Phase2 ,
162     Avg_Grade_Phase3 ,
163     Avg_Avg_GradeGroup_Phase3
164 )
165 --Retrieval Data Statement from temporary table
166 SELECT
167     Student_Term_Class_Enrollment_Key , Class_Term_Key , Student_Key
        , Class_NoOfStudent , GradingSystem ,
168     Avg_Grade_Phase1 ,
169     Avg_Avg_GradeGroup_Phase1 ,
170     Avg_Grade_Phase2 ,
171     Avg_Avg_GradeGroup_Phase2 ,
172     Avg_Grade_Phase3 ,
173     Avg_Avg_GradeGroup_Phase3
174 FROM tbl_avg_grade_All
```

Listing B.19: Inserting and Retrieval Scripting for Aggregation table: dbo.Aggregate_Brightspace_AllGrades

B.0.4 Analysis Layer - Retrieval Queries for Machine Learning Projects

```
1  SELECT
2  Dim_Student_Key
3  ,Class_StartTerm
4  ,Class_TermCode
5  ,Student_Term_Number
6  ,Student_Gender
7  ,Start_Residency_Status
8  ,From_Country_Name_EN
9  ,Student_StartTerm
10 ,Modality_Description
11 ,Course_Code
12 ,Subject_Name_EN
13 ,Class_Language
```

```
,Current_Age
,Program_Credits_Required
,Program_Normal_Completion
,Diploma_Description_EN
,Study_Field
,Years_of_Education
,Form_of_Study
,Academic_Load
,Total_Units_Term
,Current_Year_of_Study
,Current_Residency
,GradeGroup
,Course_NoOfAttempt
,[Cumm_PreviousTermEarnedUnitRates]
,[Cumm_PreviousTermPercentComplete]
,[Cumm_PreviousTermGPA]
,[Cumm_PreviousTermPassedRates]
,[Cumm_PreviousTermAtRiskRates]
,[Cumm_PreviousTermDroppedRates]
,[Course_PreviousTerm_AVGDroppedRates]
  FROM [dbo].[Aggregate_Student_Class_Enrollment]
  where --All class start since Fall 2021 to Fall 2024
( Class_TermCode >= '2219') AND ( Class_TermCode < '2249')
```

Listing B.20: Data Retrieval Scripting for Python programming

Appendix C

Prediction Models and Pipeline Codes

C.1 Phase1: Data Exploration

The complete Python implementation and the corresponding outputs for each stage of this study are maintained in a private GitHub repository. Access to the repository, including the generated result files, is available at: https://github.com/mytu007/uOttawa_ThesisProject/blob/main/Expoloration_AllPhases.ipynb.

Listing C.1: Python Scriptings: Data Exploration

```
# %% [markdown]
# # 1. INSTALLATION AND DATA IMPORT

# %%
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
# for additional visual style
sns.set_context('notebook')
sns.set_style('white')

# Load the data
```

```
data = pd.read_excel('/Volumes/Documents/TuLam/VisualStudioCode/ThesisProject/Phase3/data_allphases.xlsx')
data['IsTestData']= data['Class_TermCode'].apply(lambda x: 1 if x == 2249 else 0)

#DROP THE COLUMNS THAT ARE NOT NEEDED
columns=[
'IsTestData',
'Dim_Student_Key',
'StudentRegistered_Institution_Key',
'StudentRegistered_Registered_Faculty_Service_Key',
'StudentRegistered_Registered_Program_Key',
'StudentRegistered_Registered_Plan_Key',
'StudentRegistered_Program_Name_EN',
'StudentRegistered_Plan_Language',
'StudentRegistered_Academic_Career_Key',
'Subject_Key',
'Instruction_Mode',
'Course_Name_EN',
'Subject_Key',
'Degree_Name_Formal_EN',
'Years_of_Education',
'Total_Units_Cumulative',
'CourseName',
'Course_Grading_Basis',
'Course_Grading_Basis_Override',
'Course_Grade_Earn_Credit',
'Course_Grade_Include_In_GPA',
'Course_Grade_Units_Attempted',
'Course_Grade_Points',
'Course_Grade_Points_Per_Unit',
'Course_Grade_Official',
'Class_Term_Enrollment_DropDate',
'Class_Term_Enrollment_Status',
'GradeGroup',
'Class_TermCode',
'Student_StartTerm',
'Class_StartTerm',
'Class_Term_Key',
#Only one value because of the filter
```

```
'Program_Credits_Required',
'Program_Normal_Completion',
#Remove data from other phases
# 'Avg_Avg_GradeGroup_Phase1_Adjusted',
# 'Avg_Avg_GradeGroup_Phase2_Adjusted',
# 'Avg_Grade_Phase1_Adjusted',
# 'Avg_Grade_Phase2_Adjusted',
'Avg_Avg_GradeGroup_Phase1',
'Avg_Grade_Phase1',
'Avg_Avg_GradeGroup_Phase2',
'Avg_Grade_Phase2',
]

df = data.drop(columns, axis=1)

# Change specified columns' values to value*100 for better
   visualization
columns_to_multiply = [
    'Cumm_PreviousTermEarnedUnitRates',
    'Cumm_PreviousTermPercentComplete',
    'Cumm_PreviousTermPassedRates',
    'Cumm_PreviousTermAtRiskRates',
    'Cumm_PreviousTermDroppedRates',
    'Course_PreviousTerm_AVGDroppedRates',
    'Avg_Grade_Phase3',
    'Avg_Avg_GradeGroup_Phase3'
]

df[columns_to_multiply] = df[columns_to_multiply] * 100
# Set the index to 'Student_Term_Class_Enrollment_Key'
df.set_index('Student_Term_Class_Enrollment_Key', inplace=True)




# %% [markdown]
# # 2. UNDERSTANDING DATA

# %% [markdown]
#
```

```
# ## 2.1 Data Shapes & Categories

# %%
# 1. Understand the Dataset
print("Dataset Shape:", df.shape)
print("Column Data Types:\n", df.dtypes)
display(df.head())

# %% [markdown]
# ## 2.2 Basic Statistics

# %% [markdown]
# ### 2.2.1 Numeric Features
# - **Count**: Number of non-null entries
# - **Standard Deviation (std)**: Measure of the dispersion of the data
# - **Mean**: Average value
# - **Minimum (min)**: Smallest value
# - **Maximum (max)**: Largest value
# - **25% (Q1)**: First quartile
# - **75% (Q3)**: Third quartile

# %%

print("Basic Statistics:\n", df.describe())



# %% [markdown]
#
# ### 2.2.2 Categorical Features
# - **Count**: Number of non-null entries
# - **Unique**: Number of unique categories
# - **Top**: Most frequent category
# - **Top Frequencies**: Frequency of the most frequent category

# %%
#Check distribution of object features
df.describe(include='object')

```

```
# %% [markdown]
# Data numeric distribution

# %%
# Plot the distribution of the feature 'Class_NoOfStudent'
plt.figure(figsize=(10, 6))

# Histogram
sns.histplot(df['Class_NoOfStudent'], kde=True, bins=30, color='
    blue')
plt.title('Distribution of Class_NoOfStudent')
plt.xlabel('Class_NoOfStudent')
plt.ylabel('Frequency')
plt.show()

# Boxplot
plt.figure(figsize=(10, 4))
sns.boxplot(x=df['Class_NoOfStudent'], color='orange')
plt.title('Boxplot of Class_NoOfStudent')
plt.xlabel('Class_NoOfStudent')
plt.show()

# %% [markdown]
# ## 2.3 Missing/Null/Duplicate values

# %%
# 2. Check for Missing Values
print("Missing Values per Column:\n", df.isnull().sum())
print("If data is duplicated: ",df.duplicated().any())

# %% [markdown]
# # 3. DATA EXPLORATION
#

# %% [markdown]
# ## 3.1 Identify the correlation between a combination of
    independent variables (bi-variate analysis) using correlation,
    chi-squared test, t-test, z-test.

# %% [markdown]
```

```
# ### 3.1.1 corr_matrix

# %%
#7.CORRELATION ANALYSIS
#Pearson's correlation measures linear relationships between features.
corr_matrix = df.select_dtypes(include=['int64', 'float64']).corr()

# Plot the correlation matrix
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Feature Correlation Matrix")
plt.show()

def correlation_comments(corr_matrix):
    """
    Function to analyze the correlation matrix and provide comments on strong correlations.
    """
    # Analyze the correlation matrix
    strong_correlation_comments = []

    # Define threshold for strong correlations
    strong_threshold = 0.7

    # Iterate through the correlation matrix
    for col in corr_matrix.columns:
        for row in corr_matrix.index:
            if col != row:
                correlation_value = corr_matrix.loc[row, col]
                if abs(correlation_value) >= strong_threshold:
                    strong_correlation_comments.append(f"'{row}' has a strong {'positive' if correlation_value > 0 else 'negative'} correlation with '{col}' ({correlation_value:.2f}).")

    # Print the comments
    for comment in strong_correlation_comments:
```

```
202         print(comment)
203
204 correlation_comments(corr_matrix)
205
206
207
208 # %% [markdown]
209 # corr_matrix => Drop these features: '
    Cumm_PreviousTermPercentComplete','Cumm_PreviousTermGPA','
    Cumm_PreviousTermPassedRates'
210
211 # %% [markdown]
212 # ### 3.1.2 cramers_v_matrix
213 # Cram r's V is a statistical measure used to assess the strength
      of association between two categorical variables. It is
    derived from the Chi-Square statistic and ranges from 0 to 1
214
215 # %%
216 from scipy.stats import chi2_contingency
217
218 # Handle cases where the denominator in the calculation of Cram r
    's V becomes zero
219 def cramers_v(x, y):
220     confusion_matrix = pd.crosstab(x, y)
221     chi2 = chi2_contingency(confusion_matrix)[0]
222     n = confusion_matrix.sum().sum()
223     phi2 = chi2 / n
224     r, k = confusion_matrix.shape
225     phi2corr = max(0, phi2 - ((k-1)*(r-1))/(n-1))
226     rcorr = r - ((r-1)**2)/(n-1)
227     kcorr = k - ((k-1)**2)/(n-1)
228     denominator = min((kcorr-1), (rcorr-1))
229     return np.sqrt(phi2corr / denominator) if denominator > 0 else
        0
230 # Select categorical columns
231 categorical_cols = df.select_dtypes(include=['object']).columns
232
233 # Calculate Cram r's V for each pair of categorical columns
234 cramers_v_matrix = pd.DataFrame(index=categorical_cols, columns=
    categorical_cols)
```

```
for col1 in categorical_cols:
    for col2 in categorical_cols:
        cramers_v_matrix.loc[col1, col2] = cramers_v(df[col1], df[
           col2])

# Convert to float
cramers_v_matrix = cramers_v_matrix.astype(float)

# Plot the Cram r's V matrix
plt.figure(figsize=(10, 8))


sns.heatmap(cramers_v_matrix, annot=True, cmap="coolwarm", fmt=".2
   f")
plt.title("Cram r's V Correlation Matrix for Categorical Features
   ")
plt.show()
correlation_comments(cramers_v_matrix)

# %% [markdown]
# => Drop these features: 'Course_Code', 'Start_Residency_Status
   ','Diploma_Description_EN'

# %% [markdown]
# ## 3.2 Identify the distribution of each variables in dataset to
    ensure data is not bias

# %% [markdown]
# ### 3.2.1 Data Imbalance?

# %%


# %%
#8.DATA IMBALANCE ANALYSIS
if 'Target' in df.columns:
    print("Class Distribution:\n", df['Target'].value_counts())
    sns.countplot(x=df['Target'])
    plt.show()

```

```
# %% [markdown]
# ### 3.2.2 Outlier analysis

# %% [markdown]
# #### 3.2.2.1 Boxplot outlier

# %%

# 4. Detect Outliers using Box Plots for numerical columns
numerical_cols = df.select_dtypes(include=['int64', 'float64']).
    columns
for col in numerical_cols:
    plt.figure(figsize=(6,4))
    sns.boxplot(x=df[col])
    plt.title(f'Boxplot of {col}')
    plt.show()



# %% [markdown]
# #### 3.2.2.2 Numeric outlier: IQR Method

# %% [markdown]
# In the code below, these values are used to calculate the
    interquartile range (IQR), which is a measure of statistical
    dispersion.
#
# - **Q1 (0.25 quantile)**: This is the value below which 25% of
    the data falls. It is also known as the 25th percentile.
# - **Q3 (0.75 quantile)**: This is the value below which 75% of
    the data falls. It is also known as the 75th percentile.
#
# The IQR is calculated as the difference between Q3 and Q1:
# \[ \text{IQR} = Q3 - Q1 \]
#
# The IQR is then used to determine the lower and upper bounds for
    detecting outliers:
# - **Lower Bound**: \( Q1 - \text{multiplierThreshold} \times \
    text{IQR} \)
```

```
# - **Upper Bound**: \( Q3 + \text{multiplierThreshold} \times \text{IQR} \)
#
# Any data points outside these bounds are considered outliers.
#

# %%
# Function to identify outliers using IQR method and print boundary values
def identify_outliers(df, col, multiplierThreshold=1.5):
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - multiplierThreshold * IQR
    upper_bound = Q3 + multiplierThreshold * IQR
    print(f"{col} - Lower Bound: {lower_bound}, Upper Bound: {upper_bound}")
    outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)]
    return outliers

# Identify and filter out outliers for each numerical column
outliers_dict = {}
bounds_dict = {}
numerical_cols = df.select_dtypes(include=['int64', 'float64']).columns
for col in numerical_cols:
    outliers = identify_outliers(df, col)
    outliers_dict[col] = outliers
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    bounds_dict[col] = {'lower_bound': lower_bound, 'upper_bound': upper_bound}
    #print(f"Outliers in {col}:\n", outliers)

# # Filter out the outliers from the dataframe
# for col in numerical_cols:
```

```
#     Q1 = df[col].quantile(0.25)
#     Q3 = df[col].quantile(0.75)
#     IQR = Q3 - Q1
#     lower_bound = Q1 - 1.5 * IQR
#     upper_bound = Q3 + 1.5 * IQR
#     df = df[(df[col] >= lower_bound) & (df[col] <= upper_bound)]

# %% [markdown]
# #### 3.2.2.3 Category Outlier: identify rare frequency

# %%
# Function to identify rare categories in a categorical feature
def identify_rare_categories(df, col, threshold=0.01):
    category_counts = df[col].value_counts(normalize=True)
    rare_categories = category_counts[category_counts < threshold
       ].index.tolist()
    print(f"{col} - Rare Categories: {rare_categories}")
    return rare_categories

# Identify rare categories for each categorical column
rare_categories_dict = {}
for col in categorical_cols:
    rare_categories = identify_rare_categories(df, col)
    rare_categories_dict[col] = rare_categories



# %% [markdown]
# # 4. PRE-PROCESSING DATA:
# Drop high correlation columns - Identify and treat missing
   values
# Identify and treat null values
# Identify and treat outlier records
# Scale data values or reduce range of variables to improve
   predictive results
# Transform independent variables into features, which can be
   understandable by machine (Feature engineering)
# 14.  Create new feature to be understandable by machine (dummy
   encoding, data split, etc.)
```

```
# 15. Reduce the number of features to improve predictive results
#

# %% [markdown]
# ## 4.1 Drop high correlation features

# %%

df=df.drop(['Cumm_PreviousTermPercentComplete',
            'Cumm_PreviousTermGPA',
            'Cumm_PreviousTermPassedRates'], axis=1)
df=df.drop(['Course_Code', 'Start_Residency_Status','
   Diploma_Description_EN'], axis=1)
print("Dataset Shape:", df.shape)
print("Column Data Types:\n", df.dtypes)


# %% [markdown]
# ## 4.2 Identify and treat missing/null values by median
# Cumm_PreviousTermEarnedUnitRates       11508
# Cumm_PreviousTermPercentComplete       11433
# Cumm_PreviousTermGPA                   11598
# Cumm_PreviousTermPassedRates           11433
# Cumm_PreviousTermAtRiskRates           11433
# Cumm_PreviousTermDroppedRates          11433
# AVG_PreviousTermNoOfRegisteredUnits    11433

# %%
# 2. Check for Missing Values
missingValue_columns = df.columns[df.isnull().any()].tolist()
print("Before treating - Columns with Missing Values: ",
   missingValue_columns)
imputer_dict = {}
for col in missingValue_columns:
    median_value = df[col].median()
    df[col]= df[col].fillna(median_value)
    imputer_dict[col] = median_value
print("After treating- Columns with Missing Values: ", df.columns[
   df.isnull().any()].tolist())

```

```
409
410
411
412 # %% [markdown]
413 # ## 4.3 Identify and treat outlier records
414
415 # %% [markdown]
416 # ### 4.3.1 Numeric Outlier
417
418 # %% [markdown]
419 # #### Remove data: Handle outlier for class_NoOfStudent
420
421 # %%
422 # Remove rows where Class_NoOfStudent is greater than the upper
       bound
423 upper_bound = bounds_dict['Class_NoOfStudent']['upper_bound']
424 df = df[(df['Class_NoOfStudent'] <= upper_bound) & (df['
       Class_NoOfStudent'] >= lower_bound)]
425
426 print("Dataset Shape after removing rows:", df.shape)
427
428 # %% [markdown]
429 # #### Cap outlier at the upper/lower bound: Other features
430
431 # %%
432 import numpy as np
433
434 #Drop those columns which have only one unique value
435
436 df=df.drop(['Course_NoOfAttempt',
437 'Cumm_PreviousTermDroppedRates',
438 'Class_Term_Units_Taken'], axis=1)
439 print("Dataset Shape:", df.shape)
440 print("Column Data Types:\n", df.dtypes)
441 #Handle outlier for class_NoOfStudent
442
443 #Handle outliers (e.g., cap them at the upper/lower bound)
444
445 numerical_cols = df.select_dtypes(include=['int64', 'float64']).
       columns
```

```
446
447 def cap_outliers(df, cols):
448     for col in cols:
449         lower_bound = bounds_dict[col]['lower_bound']
450         upper_bound = bounds_dict[col]['upper_bound']
451         df[col] = np.where(df[col] < lower_bound, lower_bound, df[
              col])
452         df[col] = np.where(df[col] > upper_bound, upper_bound, df[
              col])
453     return df
454
455 # Cap outliers in all numerical columns
456 df = cap_outliers(df, numerical_cols)
457
458
459
460 # %% [markdown]
461 # ### 4.3.2 Category outliers: replace by 'Others'
462
463 # %%
464 # Function to replace rare categories with 'Others'
465 def replace_rare_categories(df, rare_categories_dict):
466     for col, rare_categories in rare_categories_dict.items():
467         df[col] = df[col].apply(lambda x: 'Others' if x in
              rare_categories else x)
468     return df
469
470 #Re-assign rare categories_dict after dropping columns
471 rare_categories_dict = {}
472 # Re-assign categorical columns adter dropping columns
473 categorical_cols = df.select_dtypes(include=['object']).columns
474 print('Before replacing rare values:')
475 for col in categorical_cols:
476     rare_categories = identify_rare_categories(df, col)
477     rare_categories_dict[col] = rare_categories
478 # Replace rare categories in the dataframe
479 df = replace_rare_categories(df, rare_categories_dict)
480 print('After replacing rare values:')
481 for col in categorical_cols:
482     rare_categories = identify_rare_categories(df, col)
```

```
    rare_categories_dict[col] = rare_categories


# %% [markdown]
# # EXPORT TO EXCEL - PREPARE FOR MODELLING & TRAINING

# %%
# Export the dataframe to an Excel file
df.to_excel('Processed_data_Phase3.xlsx', index=False)
```

C.2 Phase2: Prediction Model

The complete Python implementation and the corresponding outputs for each stage of this study are maintained in a private GitHub repository. Access to the repository, including the generated result files, is available at: https://github.com/mytu007/uOttawa_ThesisProject/blob/main/model_data2249_Phase3.ipynb

Listing C.2: Python Scripting: Preprocessing Data and Prediction Model

```
  # %% [markdown]
# # Load and prepare data

# %%
import category_encoders as ce
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import time
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
import xgboost as xgb
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, f1_score,
   precision_score, recall_score, classification_report,
```

```
    confusion_matrix,make_scorer
from sklearn.metrics import roc_auc_score, roc_curve
from sklearn.metrics import precision_recall_curve, auc
from imblearn.over_sampling import SMOTE

sns.set_context('notebook')
sns.set_style('white')

# Load the data
data = pd.read_excel('/Volumes/Documents/TuLam/VisualStudioCode/
    ThesisProject/Phase3/Processed_data_Phase3.xlsx')

#Remove data from other phases
columns_to_remove = [
    'Avg_Avg_GradeGroup_Phase1_Adjusted',
    'Avg_Avg_GradeGroup_Phase2_Adjusted',
    'Avg_Grade_Phase1_Adjusted',
    'Avg_Grade_Phase2_Adjusted'
]

# Remove the specified columns if they exist in the DataFrame
data = data.drop(columns=[col for col in columns_to_remove if col
    in data.columns])
print(data.shape)

note_run='With Brightspace - Phase 3'
note_phase = 'Phase 3'

# %% [markdown]
# # Encoding category data

# %%

# Re-assign categorical columns after dropping columns
categorical_cols = data.select_dtypes(include=['object']).columns

# Initialize the OneHotEncoder
encoder = ce.OneHotEncoder(cols=categorical_cols.drop(['Target']),
    use_cat_names=True)

```

```
# Fit and transform the data
df_encoded = encoder.fit_transform(data)
# Verify the encoding
df_encoded['Target'] = df_encoded['Target'].map({'At-risk': 1, 'Passed': 0})

# Encode the target column
label_encoder = LabelEncoder()
df_encoded['Target'] = label_encoder.fit_transform(df_encoded['Target'])

# Display the encoded DataFrame
print("Encoded Dataset:\n", df_encoded)
print("Encoded Dataset shape:\n", df_encoded.shape)




# %% [markdown]
#
# # Scaling & Handle unbalanced dataset by SMOTE

# %%
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.utils.class_weight import compute_sample_weight

# Split the data into training and testing sets
X = df_encoded.drop('Target', axis=1)
y = df_encoded['Target']

# Add a column to indicate test data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
# Calculate scale_pos_weight for xgboost
scale_pos_weight = len(y_train[y_train == 0]) / len(y_train[y_train == 1])

#Compute sample weights for gradient boosting
sample_weights = compute_sample_weight(class_weight='balanced', y=
```

```
    y_train)
# Combine train and test data back for further processing
df_encoded = pd.concat([X_train, X_test])


# Filter the training and test data

# X_train = df_encoded[df_encoded['IsTestData'] == 0].drop('
    IsTestData', axis=1)
# y_train = X_train.pop('Target')
# print('Train size: ', X_train.shape)

# X_test = df_encoded[df_encoded['IsTestData'] == 1].drop('
    IsTestData', axis=1)
# y_test = X_test.pop('Target')
# print('Test size: ',X_test.shape)

# Apply SMOTE to the training data for regression
smote = SMOTE(random_state=42)
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train
    )

# Scale the SMOTE data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train_smote)
X_test_scaled = scaler.transform(X_test)



# %% [markdown]
# # TRAIN MODEL
# ## Initial all functions

# %% [markdown]
# ### Define parameter grids for GridSearchCV and results_df

# %%
# Define parameter grids for GridSearchCV
param_grid_rf = {
    'n_estimators': [100, 200],
```

```
    'max_depth': [10, 20, 30, None],
    'min_samples_split': [2],
    'min_samples_leaf': [4, 6, 8],
    'max_features': ['sqrt'],
    'class_weight': ['balanced']
}
param_grid_cart = {
    'class_weight': ['balanced', None],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5]
}

param_grid_svm = {
    'C': [0.1, 1, 10],
    'kernel': ['poly'],
}



param_grid_xgb = {
    'n_estimators': [100, 200],
    'max_depth': [None, 3, 6, 10],
    'learning_rate': [0.01, 0.1, 0.2],
    'scale_pos_weight': [None, scale_pos_weight]  # Wrap the
       single value in a list
}

param_grid_lr = {
    'C': [0.1, 1, 10],
    'penalty': ['l2']
}
param_grid_gbm = {
    'n_estimators': [100, 200, 300],          # Number of boosting
        stages (trees)
    'learning_rate': [0.01, 0.05, 0.1, 0.2], # Shrinks the
       contribution of each tree
    'max_depth': [5, 7, 10],    # Maximum depth of each tree

}
param_grid_gnb = {
```

```
    'var_smoothing': np.logspace(-9, 0, 10)  # Use np.logspace to generate a list of float values
}
# Define scoring metrics
scoring = {
    'accuracy': 'accuracy',
    'f1': 'f1',
    'precision': 'precision',
    'recall': 'recall',
}
results_df = pd.DataFrame(columns=['Run Date','Phase', 'Note', 'Model', 'Accuracy', 'F1 Score', 'Precision', 'Recall', 'Best Parameter', 'Training Time'])

# %% [markdown]
# ### Initial functions to:
# - Plot_learning_Curve between train & validation set
# - Plot_learning_Curve between train & test set
# - train specific models

# %%
from sklearn.model_selection import learning_curve
import matplotlib.pyplot as plt
# Function to plot learning curve
def plot_learning_curve(estimator, X, y, title="Learning Curve (Train vs Validation)", scoring="accuracy", cv=5):
    train_sizes, train_scores, test_scores = learning_curve(
        estimator, X, y, cv=cv, scoring=scoring, n_jobs=-1, train_sizes=np.linspace(0.1, 1.0, 10)
    )

    # Calculate mean and standard deviation for training and validation scores
    train_scores_mean = np.mean(train_scores, axis=1)
    test_scores_mean = np.mean(test_scores, axis=1)  # Calculate test_scores_mean
    train_losses = 1 - train_scores_mean
    validation_losses = 1 - test_scores_mean
    print("Learning curve during training model (Loss): ")
    # Plot the learning curve for losses
```

```
    plt.figure(figsize=(8, 6))
    plt.title(title + " (Loss)")
    plt.xlabel("Training Size")
    plt.ylabel("Loss")
    plt.ylim(0, 1)  # Set y-axis range from 0 to 1
    plt.grid()

    # Plot training losses
    plt.plot(train_sizes, train_losses, 'o-', color="r", label="
        Training Loss")

    # Plot validation losses
    plt.plot(train_sizes, validation_losses, 'o-', color="g",
        label="Validation Loss")

    plt.legend(loc="best")
    plt.show()


def plot_learning_curve_train_test(estimator, X_train, y_train,
    X_test, y_test, title="Learning Curve (Train vs Test)", scoring
    ="accuracy"):
    train_sizes = np.linspace(0.1, 1.0, 10)  # Training sizes from
        10% to 100%
    train_scores = []
    test_scores = []

    # Train the model on increasing subsets of the training data
    for train_size in train_sizes:
        # Get a subset of the training data
        subset_size = int(len(X_train) * train_size)
        X_train_subset = X_train[:subset_size]
        y_train_subset = y_train[:subset_size]

        # Fit the model on the subset
        estimator.fit(X_train_subset, y_train_subset)

        # Evaluate on the training subset
        y_train_pred = estimator.predict(X_train_subset)
        train_score = accuracy_score(y_train_subset, y_train_pred)
```

```
            train_scores.append(train_score)

            # Evaluate on the test set
            y_test_pred = estimator.predict(X_test)
            test_score = accuracy_score(y_test, y_test_pred)
            test_scores.append(test_score)
        print("Learning curve after training model: ")
        # Plot the learning curve
        plt.figure(figsize=(8, 6))
        plt.plot(train_sizes, train_scores, label="Training Score",
           color="red", marker='o')
        plt.plot(train_sizes, test_scores, label="Test Score", color="
           blue", marker='o')
        plt.title(title)
        plt.xlabel("Training Size (Proportion)")
        plt.ylabel(scoring.capitalize())
        plt.legend(loc="best")
        plt.ylim(0, 1)  # Set y-axis range from 0 to 1
        plt.grid()
        plt.show()
# Initialize models
models = {
    'Random Forest': (RandomForestClassifier(), param_grid_rf),
    'Decision Tree (CART)': (DecisionTreeClassifier(),
       param_grid_cart),
    'XGBoost': (xgb.XGBClassifier(eval_metric='logloss'),
       param_grid_xgb),
    'Logistic Regression': (LogisticRegression(), param_grid_lr),
    'SVM': (SVC(), param_grid_svm),
    'Gradient Boosting': (GradientBoostingClassifier(),
       param_grid_gbm),
    'Gaussian Naive Bayes': (GaussianNB(), param_grid_gnb)
}


# Function to train and evaluate a specific model
def train_model(model_name):
    if model_name not in models:
        print(f"Model {model_name} not found in models dictionary.
```

```
            ")
268         return
269
270     print(f"Training {model_name}...")
271     model, param_grid = models[model_name]
272     start_time = time.time()
273     grid_search = GridSearchCV(model, param_grid, cv=5, scoring=
            scoring, refit='recall', n_jobs=-1)
274
275     if model_name in ['Logistic Regression', 'SVM', 'Gaussian
            Naive Bayes']:
276         grid_search.fit(X_train_scaled, y_train_smote)
277         best_model = grid_search.best_estimator_
278         y_pred = best_model.predict(X_test_scaled)
279     elif model_name == 'Gradient Boosting':
280         grid_search.fit(X_train, y_train, sample_weight=
                sample_weights)
281         best_model = grid_search.best_estimator_
282         y_pred = best_model.predict(X_test)
283     else:
284         grid_search.fit(X_train, y_train)
285         best_model = grid_search.best_estimator_
286         y_pred = best_model.predict(X_test)
287
288     end_time = time.time()
289     accuracy = accuracy_score(y_test, y_pred)
290     f1 = f1_score(y_test, y_pred)
291     precision = precision_score(y_test, y_pred)
292     recall = recall_score(y_test, y_pred)
293
294     print(f"{model_name} Best Parameters: {grid_search.
            best_params_}")
295     print(f"{model_name} Accuracy: {accuracy}")
296     print(f"{model_name} F1 Score: {f1}")
297     print(f"{model_name} Precision: {precision}")
298     print(f"{model_name} Recall: {recall}")
299     print(f"{model_name} Training Time: {end_time - start_time}
            seconds")
300     print(f"{model_name} Classification Report:\n",
            classification_report(y_test, y_pred))
```

```
    print(f"{model_name} Confusion Matrix:\n", confusion_matrix(
        y_test, y_pred))

    # Save results to dataframe
    global results_df

    top_features = None
    if model_name in ['Random Forest', 'Decision Tree (CART)', '
        XGBoost', 'Gradient Boosting']:
        feature_importances = best_model.feature_importances_
        feature_names = X_train.columns
        importance_df = pd.DataFrame({'Feature': feature_names, '
            Importance': feature_importances})
        importance_df = importance_df.sort_values(by='Importance',
            ascending=False)
        top_features = ', '.join(importance_df['Feature'].head(5).
            tolist())
    elif model_name == 'Logistic Regression':
        feature_importances = best_model.coef_[0]
        feature_names = X_train.columns
        importance_df = pd.DataFrame({'Feature': feature_names, '
            Importance': feature_importances})
        importance_df = importance_df.sort_values(by='Importance',
            ascending=False)
        top_features = ', '.join(importance_df['Feature'].head(5).
            tolist())
    elif model_name == 'SVM' and 'linear' in best_model.kernel:
        feature_importances = best_model.coef_[0]
        feature_names = df_encoded.columns
        importance_df = pd.DataFrame({'Feature': feature_names, '
            Importance': feature_importances})
        importance_df = importance_df.sort_values(by='Importance',
            ascending=False)
        top_features = ', '.join(importance_df['Feature'].head(5).
            tolist())
    else:
        top_features = 'N/A'

    results_df = pd.concat([results_df, pd.DataFrame({
        'Run Date': [pd.Timestamp.now()],
```

```
330             'Phase': [note_phase],
331             'Note': [note_run],
332             'Model': [model_name],
333             'Accuracy': [accuracy],
334             'F1 Score': [f1],
335             'Precision': [precision],
336             'Recall': [recall],
337             'Training Time': [end_time - start_time],
338             'Best Parameter': [grid_search.best_params_],
339             'Top Features': [top_features]
340         })], ignore_index=True)
341 
342 
343         # Save the trained model and its predictions
344         models[model_name] = {
345             'model': best_model,
346             'y_pred': y_pred,
347             'y_train_pred': best_model.predict(X_train),
348             'train_accuracy': accuracy_score(y_train, best_model.
                predict(X_train)),
349             'test_accuracy': accuracy,
350             'train_f1': f1_score(y_train, best_model.predict(X_train),
                 average='weighted'),
351             'test_f1': f1,
352             'precision': precision,
353             'recall': recall,
354             'confusion_matrix': confusion_matrix(y_test, y_pred),
355             'classification_report': classification_report(y_test,
                y_pred, output_dict=True),
356             'best_params': grid_search.best_params_,
357             'training_time': end_time - start_time,
358             'Run Date': [pd.Timestamp.now()]
359         }
360 
361         # Feature importance
362         if model_name in ['Random Forest', 'Decision Tree (CART)', '
            XGBoost', 'Gradient Boosting']:
363             feature_importances = best_model.feature_importances_
364             feature_names = X_train.columns
365             importance_df = pd.DataFrame({'Feature': feature_names, '
```

```
            Importance': feature_importances})
        importance_df = importance_df.sort_values(by='Importance',
            ascending=False)
        print(f"{model_name} Feature Importances:\n",
            importance_df)
    elif model_name == 'Logistic Regression':
        feature_importances = best_model.coef_[0]
        feature_names = X_train.columns
        importance_df = pd.DataFrame({'Feature': feature_names, '
            Importance': feature_importances})
        importance_df = importance_df.sort_values(by='Importance',
            ascending=False)
        print(f"{model_name} Feature Importances:\n",
            importance_df)
    elif model_name == 'SVM' and 'linear' in best_model.kernel:
        feature_importances = best_model.coef_[0]
        feature_names = df_encoded.columns
        importance_df = pd.DataFrame({'Feature': feature_names, '
            Importance': feature_importances})
        importance_df = importance_df.sort_values(by='Importance',
            ascending=False)
        print(f"{model_name} Feature Importances:\n",
            importance_df)

    # Evaluate underfitting/overfitting
    plot_learning_curve(best_model, X_train, y_train, title=f"
        Learning Curve for {model_name}", scoring="accuracy", cv=5)
    plot_learning_curve_train_test(best_model, X_train, y_train,
        X_test, y_test, title=f"Learning Curve (Train vs Test) for
        {model_name}", scoring="accuracy")



# %% [markdown]
# ## Random Forest

# %%

train_model('Random Forest')

```

```
394
395
396 # %% [markdown]
397 # ## Decision Tree (CART)
398
399 # %%
400 train_model('Decision Tree (CART)')
401
402
403 # %% [markdown]
404 # ## Gradient Boosting from Sklearn library
405
406 # %%
407 train_model('Gradient Boosting')
408
409 # %% [markdown]
410 # ## Gradient Boosting from XGBoost library
411
412 # %%
413 train_model('XGBoost')
414
415
416 # %% [markdown]
417 # ## Logistic Regression
418
419 # %%
420 train_model('Logistic Regression')
421
422
423 # %% [markdown]
424 # ## Support Vector Machine (SVM)
425
426 # %%
427 train_model('SVM')
428
429
430 # %% [markdown]
431 # ## Gaussian Naive Bayes
432
433 # %%
```

```
434
435 train_model('Gaussian Naive Bayes')
436
437 # %% [markdown]
438 # # EXPORT RESULTS
439
440 # %%
441 from openpyxl import load_workbook
442 import pandas as pd
443
444 notebook_path ="/Users/tumeoo/Library/CloudStorage/OneDrive-
       UniversityofOttawa/Thesis Project/Model Results/
       Result_AllPhases.xlsx"
445
446 # Append the DataFrame to an existing Excel file
447 try:
448     # Try to load the existing workbook
449     book = load_workbook(notebook_path)
450     with pd.ExcelWriter(notebook_path, engine="openpyxl", mode="a"
          , if_sheet_exists="overlay") as writer:
451         # Assign the loaded workbook to the writer
452         writer._book = book
453         writer._sheets = {ws.title: ws for ws in book.worksheets}
454
455         # Get the last row in the existing sheet
456         if "Sheet1" in writer.sheets:
457             startrow = writer.sheets["Sheet1"].max_row
458         else:
459             startrow = 0
460
461         # Write the new data starting from the next row
462         results_df.to_excel(writer, index=False, sheet_name="
            Sheet1", startrow=startrow, header=startrow == 0)
463 except FileNotFoundError:
464     # If the file does not exist, create a new one
465     with pd.ExcelWriter(notebook_path, engine="openpyxl") as
          writer:
466         results_df.to_excel(writer, index=False, sheet_name="
            Sheet1")
```