DEVELOPING TEMPORAL MACHINE LEARNING APPROACHES TO SUPPORT MODELING, EXPLAINING, AND SENSEMAKING OF ACADEMIC SUCCESS AND RISK OF UNDERGRADUATE STUDENTS

by

Nasheen Nur

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2021

Approved by:

____

Dr. Wenwen Dou

____

Dr. Mary Lou Maher

____

Dr. Mohsen Dorodchi

____

Dr. Xi Niu

____

Dr. Kim Buch

ABSTRACT

NASHEEN NUR. DEVELOPING TEMPORAL MACHINE LEARNING APPROACHES TO SUPPORT MODELING, EXPLAINING, AND SENSEMAKING OF ACADEMIC SUCCESS AND RISK OF UNDERGRADUATE STUDENTS. (Under the direction of DR. WENWEN DOU)

The main goal of learning analytics and early detection systems is to extract knowledge from student data to understand students' trends of activities towards success and risk and design intervention methods to improve learning performance and experience. However, many factors contribute to the challenge of designing and building effective learning analytics systems. Because of the complexity of heterogeneous student data, models designed to analyze it frequently neglect temporal correlations in the interest of convenience. Moreover, the performance descriptions gained from the student data model or prediction results from the analytical models do not always help explain the "why" and "how" behind it. Furthermore, domain specialists cannot participate in the knowledge discovery process since it necessitates significant data science abilities, and an analytical model appears as a black box to them.

This research aims to develop analytical models that enable domain experts to study their students' performance behavior and explore trustworthy sources of information with the help of explanations on the analytics. The work demonstrates various approaches to using the temporal aspect of heterogeneous student data to build analytical models: weighted network analysis, unsupervised cluster analysis, and recurrent neural network analytics. The description, implementation process, and findings of each method are presented as technical contributions to the temporal analysis of student data. All these analytical models highlight the complexity of heterogeneous-temporal data, model building, decision-making tasks, and the need for a more in-depth focus on visual information of analytics with state-of-art explainable AI tools and techniques.

This dissertation work underscores a need for developing a robust way to integrate the possibilities inherent within each approach. To achieve this goal, a comprehensive yet flexible and empirical framework named FIND is presented to support the design and development of analytical models to extract meaningful insights about students' academic performance and identify early actionable interventions to improve the learning experience. The framework is illustrated on three applications (e.g., student network model, unsupervised clustering model, and recurrent neural network analytics) to demonstrate the value of this framework in addressing the challenges of using student data for learning analytics. These applications present vast opportunities to benefit students' learning experience by implementing flexible educational data representations, fitting different predictive models, and extracting insights for designing prescriptive analytics and building strategies to overcome perceived limitations.

An academic institution's culture drives its ability to accept, leverage, and deploy predictive and prescriptive analytics to enhance the workflow of maximizing pedagogical outcomes. This dissertation may aid in developing or refining a set of design standards for learning analytics systems.

ACKNOWLEDGEMENTS

I would like to thank all my collaborators and committee members for their curiosity, expertise, feedback, and support that brought this dissertation to fulfillment.

I would like to thank my adviser, Dr. Wenwen Dou, for her assistance in expanding my knowledge of machine learning to explainable AI and visual analytics, proactive guidance, and commitment to fulfilling the milestones of my Ph.D. research. I want to thank Dr. Dorodchi for his mentorship and guidance throughout my Ph.D. career, especially in CS education. I would also like to thank my other dissertation committee members, Dr. Mary Lou Maher and Dr. Kim Buch, for their supportive feedback.

I would like to thank Dr. Audrey Rorrer for providing a better understanding of student data and Douglas Samson for obtaining prompt access to the student data. My special thanks to Dr. Noseong Park for guiding me on the network analytics research. I am also thankful to all the past and present group members of the learning analytics and visualization lab for the brainstorming, white-board discussions, feedback, and friendship. Finally, I praise Aileen Benedict for her contemplations and nourishing conversations.

Most importantly, I genuinely appreciate my husband, Mohiuddin Ahmed, for his composure, love, and support during my doctoral study. I couldn't have done this without my wonderful family and friends' support. They know who they are, and I will forever be indebted to them. I would love to mention the support from my mother, Soheli Akter, for her love and being a role model of a strong woman, my father Nur-e-Alam for his wise and bold guidance, my sister Nashtatin Nur and brother-in-law Md. Rafiul Faisal for their solidarity.

DEDICATION

To my only nephew, Nehan, who has brought joy, love and laughter into my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

In the educational sector, the recent surge of advances and adoption of data-driven models(Educational Data Mining) has resulted in the large-scale use of AI approaches across learning analytics applications. Learning analytics (LA) is a knowledge discovery process that combines data science, theory, and design to measure and analyze the data related to learners to improve their performance [12, 13]. Over the past decade, the increase in data volume in this sector emphasizes the need for specialized and rigorous analytical techniques to interpret student data better. Poor implementation of these analytical techniques and failure to collect an adequate amount of institutional data to support these analyses can lead to flawed interpretation and decision-making and ultimately lead to the abandonment of the techniques. Learning analytics research has focused on learning analytics in diverse institutions rather than evaluating a broad framework of its effects on the higher education sector.

This dissertation presents innovative learning analytics and educational data mining techniques to improve students' performance and learning experience. In particular, this dissertation addresses how different analytical methods can quantify the learning experience on the same student data. A framework named FIND is presented as a structure to explain our objectives with three separate analytical models. This framework comprises three main building blocks: 1 - student data engineering, 2 - learning model generation, 3 - discovering insights reflection. The three learning models are weighted student network analysis, unsupervised cluster analysis, and recurrent neural network on student data.

The findings from this learning analytics research can create interventional measures to help educational stakeholders guide their students. In this dissertation, instructors, students, educational policymakers, and education researchers are all possible stakeholders who can benefit from the knowledge collected through learning analytics and educational data mining to enhance their teaching and students' learning experiences.

1.1 Motivation

Through this dissertation work, two primary challenges were encountered in building effective analytical systems on student data. Firstly, dealing with the big and sparse heterogeneous student data with class imbalance is one of the biggest challenges. Student data is not consistently populated for every student; data can be incomplete, inconsistent, entered incorrectly, and infrequently. Apart from the problem of sparsity and incompleteness, the data demonstrates heterogeneity due to several reasons including the frequent changes in program and curriculum, frequent course withdrawal, and change of majors. Furthermore, depending on how student risk is defined, student data is likely biased toward successful students, with a significantly lower number of students at risk. Academic success or risk can be defined in many ways by different institutions. It may also vary within an institution with different disciplines establishing their own criteria for student success. For example, if the risk factor was defined as a 3.00 CGPA, it might not be very reasonable since many people would consider that CGPA threshold too high; however, it might divide the class labels of being successful and risky into a 50:50 ratio. An 85:15 proportion of successful students and students at -risk is noticed with the CGPA threshold set to 2.00 on our student data. Moreover, the ratio also varies when the ways of defining academic risk factors are changed. For example, the ratio changes when the risk factor is set to the "time to graduation" rather than defining the risk with "CGPA."

Identifying the best model and building an effective analytical system to achieve the best analytical performance on student data is another big challenge. There are dozens of ready-to-use machine learning and deep learning algorithms to tackle predictive analytics. All of these techniques are used to solve supervised classification or regression analysis issues using a large volume of labeled data. Unsupervised machine learning and deep learning algorithms are less applicable in production due to the difficulties of analyzing unlabeled data. However, researchers sometimes apply unsupervised deep learning or clustering algorithms to discover complex patterns in unlabelled data [14, 15]. Supervised learning could perform the task of prediction by having sufficient labeled data. In contrast, unsupervised learning could demonstrate specific non-trivial dependencies or even complex patterns that lead to deeper insights [15]. In addition to the diversity of applications of different analytical approaches, we also experience that both the heterogeneity and complex structure of the student data contribute to the challenge of selecting a proper analytical model. For example, student data consists of continuous and discrete numerical features, categorical features, and sometimes unstructured textual data. This data is also temporal and varies significantly over time.

Moreover, during the process of selecting appropriate features to formulate the best performance metrics, domain experts need to be involved. Domain experts are the educational leaders, teachers, and academic advisers who can evaluate the students' performance as well as the efficacy of the tools to analyze the performance. The necessity of involvement of the domain experts is a perfect example of accuracy does not equal insights - when machine learning cannot generate insightful outcomes unless there is involvement of domain knowledge from the domain experts. Domain experts know the data, the context and the students they interact with; such wealth of knowledge is critical for data analysis and decision making.

The implications of the challenges mentioned above include:

1. Many features about student academic performance can be sparsely populated.

2. Most Predictive models expect a single data type.

3. Predictive models demand a consistent feature vector or data structure.

4. Temporal data requires specialized AI models.

5. The volume of data on at-risk students is insufficient for building accurate predictive models.

6. The involvement of the domain experts is crucial to validate the feature selection and sensemaking process on the analytical models.

After thoroughly researching the existing challenges, two main gaps are distilled from the current learning analytic approaches and the existing learning analytics systems. The first research gap is that most methods do not use temporal models for predicting students' at-risk [16, 17, 18]. However, the bottleneck in solving this challenge is the heterogeneity of the students' data hinders integrating different data sources for modeling, and the sparsity of some features could be necessary for some specific population [3]. The second research gap focuses on the "lack of explainability" of predictive results and models in the learning analytics system. The practice of interpretable machine learning has not yet been adopted by learning analytics research on a large scale. Often, these analytical tools lack the explanations behind a particular predictive result or a visual report on the working mechanism of the features in an analytical model. Conati *et al.* [19] is the first person who emphasizes the need for eXplainable AI (XAI) in open learner models. The authors believe XAI will help policymakers and academic advisers to understand their students better.

1.2 Thesis Statement and Research Questions

> **Thesis Statement:** Temporal analysis and providing explainability of machine learning models on large-scale heterogeneous student data demonstrate a better prediction and understanding of students' behavioral patterns towards their performance than black-box or non-temporal machine learning models.

This thesis addresses the following research questions for three different building blocks of our framework:

- Student Data Engineering
  - **RQ1.1:**Leveraging the temporal nature of the student data, what do the different student data models capture?
  - **RQ1.2:**What is the benefit of addressing heterogeneity of high-volume student data?
- Learning Models Generation
  - **RQ2:**How do different explainable machine learning methods help produce better models of student success and risk? (comparison of methods)
- Insight Discovery
  - **RQ3:**How does each model support comprehending students' performance and effectively planning interventions? How may the insights gained from these models aid domain experts in obtaining actionable knowledge?)

The research questions **RQ 1.1-1.2** are connected to chapter 4, which discusses several student data representations. **RQ2** and **RQ3** are related to chapter 5 and chapter 6, respectively. All the research questions are answered and reiterated at the beginning of chapter 7.

This dissertation aims to frame and demonstrate the importance of student data representations and data preparation on the performance of machine learning algorithms. The research demonstrates an effort to compare different machine learning models in education on the same student data and understand how these different models can affect the interpretation of students' performance via model explanations and visual analytics.

1.3 Thesis Outline

The remainder of this thesis is organized as follows:

Chapter 2 reviews the background information crucial to understanding this thesis, including a brief discussion of current learning analytics techniques and tools. Relevant knowledge discovery or educational data mining approaches are also introduced. This chapter focuses on the existing challenges in current analytical practices and discusses the gaps in the decision-making efforts from the predictive outcome of the prediction models.

In chapter 3, a unifying framework named FIND is presented to consolidate and abstract the stages and issues for designing and refining learning analytics systems. The building blocks of the three learning models - weighted student network model, unsupervised clustering model and recurrent neural network on student data are discussed as a demonstration of the value of this framework in addressing the challenges of using student data for learning analytics from chapter 4 to chapter 6.

To benefit from the explanations as a feedback mechanism, it is critical to understand the interaction between different data formats and analytical models. In chapter 4, different data representations and feature vectors on the same student data built for different analytical approaches are discussed.

In chapter 5, we explore different analytical models and how each of them is contributing to understanding the students and their performance. Chapter 6 demonstrates how the visualization of students' collaboration networks, model exploration with explainability tools deepens the understanding and improvement of individual models.

Chapter 7 summarizes the contributions, findings, and limitations. Finally, some future directions on this research are mentioned in this chapter.

1.4 List of Contributions

This dissertation incorporates newer methods in the area of learning analytics to understand student risk/success and address the aforementioned research gaps by:

1. **Developing three analytical models to understand how new analytical techniques and tools help to gain deeper insights on students' risks and success:** To achieve this goal, three different machine learning and data models (weighted network analytics, unsupervised clustering and recurrent neural network based analysis) were developed. Several machine learning algorithms on various student data models to better understand students' overall performance and determine the factors that influence the students' performance are explored and implemented throughout the process. Even though the objective is the same, different machine learning algorithms demand different types of inputs, i.e. prediction and categorization of successful and at-risk students is a common target for different predictive analysis. Different preprocessing techniques were applied on the same student data to prepare different inputs of student data models for the different machine learning algorithms.

2. **Incorporating explainable AI and visual analytical results to assist the understanding of the predictive models:** The segregated clusters of the unsupervised clustering on engineered-temporal features were explored to identify the behavioral patterns of different student cohorts. In addition, two different state-of-art explainable AI tools are discussed on the recurrent neural network models to demonstrate the efficacy of interpretability for model improvement and actionable decision-making.

3. **Designing a prototype for an interactive dashboard to involve domain experts in the decision making process:** In this prototype, a data model, an aggregate analytics view with visual and tabular explanations and a view to

understand students individually were incorporated with different attributes. Furthermore, the originality of this research demonstrates the significance of explainable AI for educators and policymakers to comprehend model predictions. Student advisors participated in evaluating and providing feedback on the prototype.

4. **Developing an empirical framework named FIND to structure the whole learning analytics process into actionable insight generation:** A generalizable framework to demonstrate how the three different learning models help the domain experts and educational policymakers to identify actionable insights is presented in this dissertation.

Here is a summary of the contributions of this dissertation:

1. This dissertation develops three temporal machine learning models to predict and analyze students' behavior towards their performance. The main contribution of this dissertation is to extract different insights and actionable knowledge from those models on the same dataset and features. Each model demonstrates separate things depending on the data model and how the machine learning models are formulated.

2. The weighted student network model extracted from an offline student database is an innovative approach to learning analytics.

3. This dissertation presents an innovative approach by using XAI in an interactive dashboard to involve the domain experts in proper model training and understanding the students better.

4. This dissertation first uses recurrent neural networks for degree analytics and demonstrates the models' efficacy with SHAP explanations by showing the time-

wise (semester-wise) performance for both regression and classification problems.

5. This dissertation presents a framework to compare the innovations across three learning models from data representations to insight generation to help the domain experts understand students better.

CHAPTER 2: Background

In computer education research, learning analytics (LA) is the process of evaluating data about learners to improve their educational attainment, learning experience, and operational efficiency [20, 21].

Learning analytics is defined by Sclater *et al.* as "the measurement, collection, analysis, and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs" [22]. Over recent years, characterizing the educational experience became complex and more complicated since it requires interdisciplinary expertise, starting from data experts to educational leaders. LA is now a multidisciplinary knowledge discovery process that demands to include data science, theory, and design to achieve an optimized and efficient learning experience to uncover patterns in students' success or failure outcomes [20, 12, 13]. Researchers in this field have begun to incorporate newer methods to gain more insight into learning programs and tools [23]. The operational research of learning analytics requires knowledge of educational data mining and analytics and interactive recommender systems for personalized adaptive learning [24].

At first, numerous educational choices by academic policymakers or domain experts such as faculty, staff, and administrators have been based on instinct. However, decision-making can be improved when grounded on data, actualities, and factual examination [22]. As datasets on learners proceed to develop and become more promptly accessible, the potential for learning analytics grows too. This development of enormous information in big data drives the improvement of the tools and strategies of learning analytics [22, 25]. Moreover, it should not fundamentally substitute human qualities such as engagement, ability, and judgment; perhaps they should be supplemented by exposure methods where appropriate [22, 25]. This advance emphasizes the significance of learning analytics in education.

This chapter will discuss the data science and knowledge discovery aspects of current state-of-the-art learning analytics approaches and tools. The discussion will center on learning analytics workflows and knowledge discovery processes, hurdles in preprocessing large amounts of heterogeneous student data, addressing the research gap of implementing different machine learning approaches on the student data model, and defining sensemaking research interpretation on predictive results. In this thesis, improving overall learning experiences is emphasized rather than solely focusing on performance.

2.1 The Overall Workflow of Learning Analytics

Learning analysts and researchers portray the workflow of extensive learning analytics processes differently. These workflows tend to direct projects from basic information to experiences and insights (the method of KDD) and tend to vary on data examination strategies, including stakeholders and pedagogical practitioners [26]. The researchers agree that the KDD method profoundly takes after the manner of any knowledge discovery process. Chatti *et al.*depict the significant steps of the learning analytics process as (1) data gathering and preprocessing, (2) analytics and intervention, and (3) post-processing [24]. They claim that the education data is the basis of the learning analytics (LA) process [24]. Data gathering and preprocessing are fundamental to perform any analytics on the data related to the learners. Preprocessing step reconstructs the input data into the essential format to be input into a particular LA model. This step mainly incorporates "data cleaning, integration, data transformation, data reduction, data modeling, user and session identification, and path completion" [24]. Based on the preprocessed data, analytics and intervention procedures are used to study the learners' performance, uncover insights, and improve the learner experience. Chatti *et al.* further mention that the analytics step incorporates the study and visualization of data and includes monitoring, analysis, prediction, mediation, evaluation, and reflection. Post-processing at that point consists of the steps to make strides in the analytics cycle ceaselessly. The process could incorporate unused or new information, refining the current dataset, or characterizing new features for the following cycle.

According to Han *et al.*, the workflow of the analytic learning process comprises four crucial steps: 1) data cleaning, 2) feature extraction, 3) data mining, and 4) evaluation to learn compelling insights from the student data model [2]. Per the experience gathered throughout this dissertation in learning analytics research, the fifth step is considered a post-action or post-processing step called "intervention" according to the language of Computer Science (CS) education. This section will describe the five stages of the learning analytics process (see Figure 2.1).

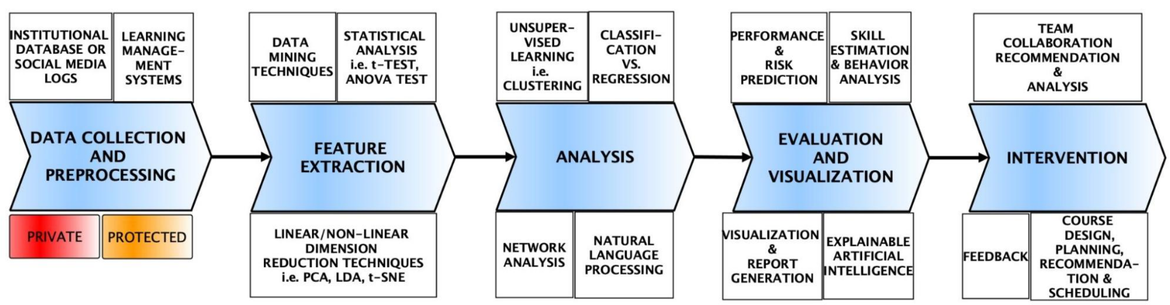


Figure 2.1: The Overall Workflow of Learning Analytics encouraged by Han *et al.* [2]

2.1.1 Data Collection and Preprocessing

In general, data is collected for learning analytics from a broad range of technologies and contexts in institutional settings. Student Information Systems (SIS), Intelligent Tutoring Systems (ITS), Learning Management Systems (LMS), Personal Learning Environments (PLE), social media logs, survey and interview questionnaires, and game-based and simulation-based learning environments (PLE) are examples of data sources.

With the advancement of open-source projects and Internet access via small mobile devices, learners are yielding to online learning activities. Therefore, learning management tools have to deal with a lot of data originating from learners' activities and growing enrollments. Learners' data are either protected or private; they are not openly accessible. Due to the enormous amount of data produced by the overall system, learning analytical research shifts the focus from traditional data gathering and processing approaches to "Big Data" techniques [27]. Student tracking data straight from learning management tools consists of logs from various events, interactions, and academic performance, which need particular attention to collect, and manipulate. For example, Blikstein [28] presented NetLogo, a framework to observe students' behavior while writing codes in their programming assignments. The tool captures large quantities of events such as keystrokes, button clicks, and changes. Usual RDBMS(Relational Database Management System) tools are insufficient to save and prepare the vast amount of data coming from education [23]. MongoDB, Hadoop, MapReduce, Orange, Weka are some open source ETL (Extract, Transform, and Load) tools to help store and process the data [29]. Some generic to specific data preprocessing tasks are necessary to help transform the first bulk of data into proper analytic research. Preprocessing techniques include data sanitation, user anonymization, session and transaction identification, path completion, data transformation, enrichment, and reduction [30].

2.1.2 Feature Extraction

Collected and preprocessed data for storage purposes is converted to a specific student data model, which accommodates the features for a particular analytical approach. For instance, Nur *et al.* extricate different academic features, including GPA, high school records, and academic progression features (e.g., number of course withdrawals, academic standing, credits attempted and earned) to anticipate on time and delayed graduation in an education institution [1]. The authors in this publication build student network models with data from students other than interactions reported in learning management systems or social media logs. They extracted different network features from the weighted student network model and a fixed point feature in addition to the classical academic features.

Different data mining techniques, statistical methods, i.e., t-test, ANOVA analysis, linear or non-linear dimensionality reduction algorithms, i.e., nearest neighbor analysis, are used to perform feature extraction procedures. The extracted features are engineered features or basic features directly retrieved from the raw data, entirely dependent on the individual analytical approach. Different performance features, student activities, demographics, and background information are the most commonly extracted features.

2.1.3 Analysis

In this step, researchers apply a statistical or machine learning method or a set of analytical or data mining strategies to gauge students' performance or determine experiences and insights from the extricated features. Some techniques for analysis in learning analytics or educational data mining are statistical analysis, regression, classification, clustering, text mining, nearest neighbor analysis, network analysis, natural language processing, and model discovery [23]. A supervised machine learning method, for example, can be used to create a predictive model for classifying students based on course performance data. Later, this supervised model can be used on unseen students' (test data) data to predict their performance earlier in their enrollment.

Chapter 4 compares various analytical methodologies to preprocess the student data used in this dissertation. Section 2.3 and 2.4 will discuss some of the analytic methods and their challenges in the existing research.

2.1.4 Evaluation and Visualization

Evaluation of students' performance is the ultimate locus of the learning analytics workflow. The data scientists analyze students' performance, estimated risks, behavior, and skills by different analytical and pattern mining approaches. Various performance metrics such as accuracy, F1-score are assessed iteratively to optimize the performance of the analytical method.

Researchers employ visual reports and visualization approaches to detect trends, insights, and relationships in data. They assess the visual information to find out the potential of learning analytics practices to predict academic performance [31]. For example, maintaining the size of the learning community and further processing associated with it is complex, and most of the passive activities may remain unnoticed, especially in distant learning [32, 31, 33]. Visualization tools help instructors and administration to extract and visualize actor-specific and group-specific information exchange and analytics on the performance, which is usually hard to observe from the abstract level.

Several eXplainable Artificial Intelligence (XAI) tools and methodologies are now used to visually or textually comprehend the findings of prediction models. It also assists data scientists with the development and training of machine learning systems. The usage of XAI is not broadly adopted in learning analytics research. Two XAI tools are discussed on the neural network model in chapter 6 to illustrate the advantages of explainable AI in LA and EDM systems and research.

2.1.5 Intervention

Intervention in learning analytics is a set of activities that takes insights from the analytical evaluation and uses them to attain ultimate pedagogical learning goals [34]. These activities consist of giving intelligent feedback to the appropriate stakeholders like students or parents, course recommendations, course design, planning and scheduling, and suggesting new group formation to monitor changes in performance. Most of the learning management tools aim to provide automated or manual feedback to the learners in response to the engagement with the process to better their performance and collaboration [35, 16]. Learning tools integrate recommendation systems to automate course suggestions by analyzing students' past academic performance, activities, and choices [36]. Different learning analytics visualization tools help instructors to analyze student performance and recommend guidance proactively [37]. By analyzing the outcome in the evaluation step, instructors redesign the courses, schedule, and plan ahead of time. They can also suggest new groups [38] and pedagogical activities to improve performance [38, 39].

Although the ultimate purpose of the learning analytics process is to plan interventions to maximize the positive learning experience, this dissertation focuses till the evaluation step of the analytic learning process.

2.2 Categories of Learning Analytics Techniques and Tools

Many learning analytical tools and techniques are surveyed as part of this dissertation. They are categorized into two different dimensions-

1. Based on the target users, and
2. Based on the timeline of analysis on student data

Based on the target users, two types of systems are used. The first type of tools and techniques are used to improve the learners' learning experience where the target users are students. Many learning analytics systems were designed to help individual students enhance their learning experience and outcomes [40, 17]. Domain experts such as faculty advisers, practitioners, administrators use the second type of approach to address attrition in schools. Many predictive and visualization systems have been developed to address the challenge of attrition in schools [16, 41, 3]. This thesis focuses on the latter type of learning analytic techniques. Faculty leadership and advisors use the data sources hosted on this kind of "Learning Analytic Systems" to help their students achieve better performance in their academic life.

Based on the timeline of analysis on student data, two types of relevant learning analytics systems have been developed in prior research to analyze student data. It can be divided into (1) Degree Analytics and (2) Course Analytics. The first type looks at the students' progress throughout their whole program [3, 1, 42, 43], and the second type looks at students' progress on a course level or course majors [16, 44, 45]. For example, Romero *et al.* [40] used Moodle data to develop predictive models coupled with a visualization platform for advisors to better understand students' activities on a course level. Course Signals [16] is a LA tool that uses Backboard Vista to predict students at-risk and recommends interventions for the advisors to prevent attrition. Along the same lines of predictive analytics for student achievement, researchers develop the Open Academic Analytics Initiative (OAAI) [41]. They include more data sources than course grades, such as demographics and the students' interaction with course level LMS.

Table 2.1 addressed each of the types of predictive analytics mentioned above. Three papers focus on using learning analytics to study the curriculum further to improve learning experiences and results at the degree level [46, 47, 48]. Monroy *et al.*, for example, investigate how learning analytics might help enhance the curriculum by better understanding the way and to what extent the teachers and students use it [46]. Chou *et al.* focus on visualizing students' core competencies as they progress throughout the curriculum [47]. Much additional work has been done to improve student learning experiences and outcomes at the course level [40, 49, 16, 50, 51]. For example, Arnold and Pistilli created SIGNALS, a dashboard for students and instructors to view student progress [16]. Dyckhoff *et al.* emphasize the necessity of equipping instructors with the tools they need to assess their teaching approaches and make adjustments [50].

In contrast to the purpose of increasing learning experiences and outcomes, other works focus on the issue of school attrition. Two degree-level works describe strategies for predicting student graduation times so that interventions can be conducted for students who are on the verge of graduating late or not at all [3, 1]. Another work

Table 2.1: Categorizing Types of Predictive Analytics Done in LA. (Note: A paper's related work substantially built on this table is under review)

		Target Users and Expected Outcomes	
		To improve learning experience and outcomes	To address the challenge of attrition in schools
Timeline of analysis	Degree Analytics	Monroy *et al.* [46]; Chou *et al.* [47]; Snodgrass *et al.* [48];	Mahzoon *et al.* [3]; Nur *et al.* [1]; Demeter *et al.* [52]
	Course Analytics	Romero *et al.* [40]; Dorodchi *et al.* [49]; Arnold *et al.* [16]; Dyckhoff *et al.* [50]; Dietz-Uhler *et al.* [51];	Amnueypornsakul *et al.* [53];

at the degree level combines academic and financial information to predict student retention and graduation outcomes [52]. Amnueypornsakul *et al.* collect data from a MOOC (massively open online course) and train a model to predict student attrition in a course [53].

Early warning and student success are two of the most common applications of learning analytics [22]. Predictive models are constructed utilizing historical data on past students to classify or predict whether a current student is at risk or not. It depends on several aspects, including behavior or skill analysis [24, 22]. Learning analytics tools and approaches can also be classified using early alert and student success analysis, course suggestions, adaptive learning, and curricular learning [22]. The knowledge discovery and data mining techniques for these applications are discussed in section 2.3.

There are research contributions that focus on discussing a learning analytics framework in addition to specific approaches to learning analytics. Some frameworks concentrate on current data, data analysis tools, participating stakeholders or target consumers, and constraints [26, 54]. Others have pushed for stronger links between learning analytics and linked data, focusing on semantic web technologies [54, 55]. However, this strategy does not include information on the learner's history or curricular requirements [54]. Some authors focus on social learning analytics while discussing frameworks. Data mining and visualization tools, for example, have been used to visualize discussion activities [33, 54]. "Six critical dimensions of a LA framework, including stakeholders, objectives, data, instruments, internal and external constraints," according to Greller and Drachsler, are all vital for creating and implementing LA applications [26, 54]. A framework of learning analytics named FIND is presented in chapter 3, covering a broad range of learning analytics goals and methods. Rather than focusing on the breadth of LA, the approach focuses on the process and qualities of the process. It covers the various components of data engineering, model construction, and insight production and how these three processes are combined to improve learning. This dissertation demonstrates three use-cases to explore the framework.

2.3 Knowledge Discovery & Data Mining Process in Learning Analytics

This dissertation surveys the most common analytical and knowledge discovery and data mining approaches adopted by learning analytics (LA) and educational data mining (EDM.) Reviewing data features and their classifications and critical properties commonly utilized in the area is part of the preparation phase.

Research in learning analytics took off significantly when Massive Open Online Courses (MOOC) became popular. MOOCs allow for the online development of live and interactive support using LA [56]. The term "Massive Open Online Course" (MOOC) was first introduced to describe a twelve-week online course, "Connectivism and Connected Knowledge," designed by George Siemens and Stephen Downes, offered at the University of Manitoba, Canada, in the fall semester of 2008 [56]. Nowadays, MOOC is a famous form of an online education system that involves a highly diverse student population and various motives for taking the course in the absence of on-site instructors. MOOC started the journey intending to enroll "Massive" students in courses and track enormous quantities of participant activity and performance data.

With an adequate internet connection and low to free cost, this "Open" education system is becoming a center of attention in learning analytics research. MOOCs are offered via the Internet on various devices and thus reach out to learners beyond the traditional campus. Each MOOC (a single course) has a time frame with a beginning and an endpoint. Learners are provided with a coherent set of resources. An instructor organizes a regular follow-up on the sequential activities to investigate progress in the learning and sometimes offers interactive guidance.

To answer a wide range of educational research problems, EDM researchers use a number of analytical and data mining approaches such as network analysis [1, 57], text mining [58, 59, 60, 61, 62, 63], supervised machine learning [3, 64, 65], computer vision [66], curriculum mining [67, 68], and recommender systems [69, 70].

Network analytics became a leading method for studying online social interactions at an individual level (ego-centric analysis) and group-level (community-wise), as well as for studying student learning networks in interactive group support [48, 53, 46, 71]. The representations of social interactions provide a robust mechanism for studying collaborative processes in learning communities, such as knowledge formation, risk estimation, evaluation, and then suggest influence and assistance to the communities [72, 73, 26]. Researchers explore offline collaboration and academic data to avoid the challenge of social media data access, possible missing/irregular interaction logs and data points in learning management systems, and optimizing the building of network edges [1, 57].

Text mining, also known as text analytics, is an artificial intelligence (AI) tool that converts unstructured text from student conversations, comments, discussion forums, and code submissions into normalized, structured data. Then, the converted data is analyzed or used to model machine learning algorithms using natural language processing (NLP). Dorodchi *et al.* demonstrate how sentiment traits extracted from student reflections can be used to identify troubled students early on so that the instructor can intervene [49]. Besides, sentiment analysis has been used to detect bad and positive teaching practices to supplement data mining-based educational techniques.

Grade prediction, graduation outcome(success or risk) prediction are examples of predictive student analytics and supervised machine learning on student data [3, 64, 52]. Mahzoon *et al.* convert student institutional data into a sequential data model consisting of temporal nodes holding data points from sequential semesters to forecast delayed graduation in a higher education institution [3]. The first node contains information on the student's background (such as demographics, high school records, age, and so on), while the last node contains information about the student's graduation.

Innovative computer vision applications explore the relationship between the physical learning environment and student learning even though researchers reportedly proclaim hardware limitations for this data mining process [66]. Recommender systems use feedback, course and material recommendations, and assistance to help students choose the best learning paths [69, 70]. In education, these systems frequently lack a consistent review mechanism for current systems and data sets.

2.4 Challenges and Research Gaps

Numerous studies in the literature go to great lengths about developing prototypes or executing learning analytics applications and discussing the challenges and constraints that come with doing so. Data preparation and layout for the machine learning process are common hurdles in implementing LA approaches. Academic data evolves due to curriculum changes, internal and external rules, regulations, and changes to underlying information systems resulting from regulations and policies.

Two main gaps were addressed in the current learning analytic techniques and tools after a thorough survey of the existing challenges. The first research gap is that most methods do not use temporal models or temporal aspects of the features for predicting students at-risk [16, 17, 18]. Over the last decade, many ways of examining time-series data to uncover behavioral trends and patterns have been developed by the data science community. However, more needs to be done in the learning analytics community to adopt and adapt the temporal aspect of data for performance or retention analysis. The heterogeneity of the students' data hinders the integration of different data sources for modeling, leading to hesitance in using the temporal aspect of the data [3]. The second research gap addresses the "why" question that faculty leaders have in mind when analyzing students' data. Alfred and Hanan [42] emphasize the need for explanations and interpretation in open learner models. The explanations and interpretations will help the instructors and policymakers to understand the reasons behind the early performance prediction by the underlying models. This section will address both research gaps by examining research areas from data mining communities, which shows us why temporal aspects of analytics and interpretation of analytical results are two critical gaps to solve for learning analytics research.

2.4.1 Temporal Learning Analytics

Time series analysis is a part of our modern life, starting from forecasting weather to the stock market. Data scientists believe time series analysis to be superior and more dependable when there are enough data points than other methods. The developed methodology proves all the underlying assumptions of the problems [74]. When a sequential and ordered collection of observations across evenly spaced time intervals is available, time-series analysis is widely used to forecast future values [74]. Three primary assumptions in the time-series analysis are - 1) the data points are in numeric format, 2) the observations close to each other in a time series are correlated, and 3) the predicted future values depend on the currently available data values while the present values depend on the values in the past [75, 74].

Because of the complexities of diverse student data, models designed to analyze it frequently neglect temporal correlations to simplify and improve forecast accuracy.

Some researchers examine the efficacy of temporality of the student data in computer-supported collaborative learning (CSCL) and self-regulated learning (SRL) [17, 18, 76]. They emphasized time is an essential factor in analyzing students' behavior or performance. They mentioned-

- By aggregating features for the group collaborations does not indicate when or where a particular pattern occurs
- Self-regulated learning is a series of events that act differently over time and change contexts
- Students change their strategies and learning patterns over time

Many researchers argued that an extensive analysis of temporal/ sequential student data would open a new dimension for learning analytics research. They noted that the temporal features of enormous, varied student data necessitate more attention, care, and possibilities. However, the time series data assumptions indicated above do not apply to student data [76].

Student data is not necessarily numeric, according to the first assumption. Student data include textual and sometimes ordinal data in addition to its diverse nature and numeric aspects. While several types of temporal data from the learning management system (i.e., Clickstreams) have been used to model student behavioral trajectories, there has been little research on the importance of analyzing textual data, which is available in all students' records [77, 78]. Byungsoo *et al.* [79] demonstrate how incorporating textual information (mentors' reflecting remarks) increases both the prediction power of identifying at-risk students and the ability to provide insights into student course participation processes. Moreover, not much research considers the textual and temporal aspects of course numbers and course subjects, which might preserve some meaning of student progression towards their graduation. It is necessary to consider maximum temporal factors of student data to achieve robust and more vigorous analytical models.

Furthermore, unlike other time-series data, student data has no seasonality or regularities within a period, which is another aspect of student data heterogeneity. As individuals complete more credit hours or take (or are compelled to take) more difficult courses toward graduation, their progression trajectories gradually shift. Moreover, even while there may be some common trends among students with similar performance, student data is highly different from one another. The most significant advantage of using time series analysis is that we can understand the past and predict the future by using it. Moreover, a student's whole academic career is a series of events with irregular changes in strategies and performance. Therefore, ignoring those temporal effects will be like ignoring patterns and insightful meanings from their academic career. As a result, addressing temporal trends and anomalies in heterogeneous student data is required to construct a model for a better understanding of student performance.

2.4.2 Interpretable Learning Analytics

We entered a new age of AI applications where predictive analytics with machine learning plays a core technological role. The current adoptions of AI systems offer tremendous advantages in our everyday life. Nonetheless, the inability of machines to explain their choices and actions to people casts doubt on the efficacy of the treatment. The research of explainability and interpretability is relatively new. We are seeing hype in this research for the last five years. Because most machine learning models are opaque, non-intuitive, and difficult to comprehend, explainable predictive analytics on AI systems will be critical if users understand, trust, accept, and govern the systems' outcomes.

The definition of eXplainable AI (XAI) systems profoundly depends on the application areas. For example, it can focus on explaining the technical components of predictive analytics [80, 81, 82, 83] or more on exhibiting the raw data such as conversations, images, or specific trends or patterns on different media data [84, 81, 85]. The debate of emphasizing the usage of XAI systems got more vital, especially in the last three to five years due to the recent legal restrictions act on Data (for example, the GDPR [86]) and increasing usage of black-box Deep Learning models for analyzing BigData [87, 88, 89, 90].

Learning analytics dashboards that are difficult to decipher impact learners' learning experiences [91]. Alfred and Hanan [42] are two of the first education researchers who emphasize the need for explanations and interpretation in open learner models. They offer a comprehensive early warning approach for detecting and treating at-risk students using data visualizations to get diagnostic insights into the predictive model, explain students' performance and behavior, and organize treatments.

According to the authors, implementing a realistic early warning system is very challenging, and it should meet two assumptions. Firstly, the predictive model should be generalizable to switch between contexts. Secondly, the underlying mechanism of prediction should be easily interpretable by educational leaders and practitioners whose ultimate goal is to design essential and realistic interventions for the students. Conati *et al.* claim that interpretable AI systems support and understand students' cognitive, emotional, and performance behaviors, leading to optimal learning and teaching experience [19]. Some other researchers, including Conati *et al.*, emphasize that the LA systems may include one of the following three types of interactivity and interpretability - 1) scrutable, i.e., users may view the models' evaluation of relevant students specific patterns, feature importance, and underlying raw features, 2) cooperative or negotiable, where the user interacts with the system to arrive at an assessment, and 3) editable, where the user has the full power to change the model assessment and systems representation of knowledge [92, 93, 19].

This dissertation proposes a prototype for an interactive analytics tool applying the same concept on LMS. The novelty of the prototype exhibits the importance of explainable AI for advisors to understand the model predictions. Although the primary goal of interactive AI tools is to help educational leaders (e.g., the domain experts) predict students' outcomes and plan interventions, tools do not always explain the "why" behind those predicted outcomes or the results. The goal is to help the domain experts better understand their students by exploring the reasoning behind these predictions by getting involved with building the analytical models themselves. The proposed tool plans to do this by engaging them in the life-cycle of analytics by giving them the flexibility to select features to build better analytical models for predicting student success. This dissertation aims to incorporate explainable machine learning to create helpful insights by opening the black box model for analytics. Chapter 6 shows how to use two XAI tools to describe the time-series model's training and prediction outcomes. The preliminary findings provide promising insights showing how the users can get directly involved with the analytics and interpretation of the results. The screenshots of the first published version of the tool are provided in the appendix (A.2) section of the dissertation.

In summary, a human-centered design approach is used to engage faculty and advisors in developing an interactive knowledge discovery system for a better understanding of student success and students at risk. Human-centered design [also Human-centered design, as used in ISO standards] is a problem-solving strategy that involves the human perspective in all stages of the problem-solving process [52]. Observing the problem in context, brainstorming, conceptualizing, and building the solutions are all instances of human engagement in the process. The different analytical approaches discussed in chapter 5 allow faculty and advisors to interact with the data used in an analytic model, the results of the analysis, and the story of individual students.

CHAPTER 3: FIND: An Unifying Framework for Learning Analytics

Note: A paper substantially building on the chapter is under review

3.1 FIND: A Framework for Discovering Insights from Novel Student Data Models

It is contended that the challenges of complexity in data preprocessing, selecting machine learning models, and different student data representations are not adequately addressed in prior works. Most of the previous work focused on the analytical models and analysis of the outcome without thoroughly examining the representation and preprocessing techniques needed or done on their data. As the field of Learning Analytics (LA) grows, researchers have proposed several frameworks, which focus on methodologies for data analysis, involving stakeholders, and pedagogical processes [26]. Aquin *et al.* emphasize a closer relationship between data representation in LA and semantic web technologies [55]. On the other hand, his research is missing crucial information on the learner's background and curricular requirements. Various frameworks focus on social learning analytics (SLA), which provides detailed information on learning processes in real-time conversation [94]. Participants, goals, data, tools, internal and external constraints are among the six critical parts of Greller & Drachsler's [95] LA framework. When it comes to creating and implementing LA applications or systems, these dimensions are crucial. Still, holistically validated learning analytics frameworks are scarce. One of the main limitations of the existing frameworks is the missing links among data representations (e.g., prior features and processing), model generation (e.g., predictive models, pattern extraction techniques), and insights (e.g., competencies, interpretations).

Over years of experience, this dissertation agrees that there is certainly more than one way to conduct learning analytics research. One cannot simply apply a series of regression or predictive models and evaluate which is the best. Because pedagogical outcomes differ by educational institutions and the context changes within institutions, data scientists must make many decisions to develop different techniques and demonstrate additional insights. In this section, FIND for Learning Analytics, a framework for addressing data heterogeneity challenges, encouraging diverse data representations, and comparing different data mining techniques, is presented. FIND is assessed in a comparative analysis using multiple data mining techniques and multiple student data representations.

Moreover, this dissertation presents an overall workflow of the learning analytics process proposed by Han *et al.* [2] in chapter 2 (See figure 2.1). Han's framework [2] provides the answer to the "how-to" question for a learning analytics system. Even though this framework offers a step-by-step walkthrough in learning analytics systems, this workflow has different granularity than FIND. For example, this workflow shows "data collection and preprocessing" and "feature extraction" as two separate modules. However, FIND presents a holistic approach of these two modules into one module named "student data engineering" since the process is iterative and involves similar challenges within this scope. However, FIND furthermore compares the innovations, insights, and challenges in each module of the workflow.

Fig. 3.1 illustrates an overview of the presented LA framework linking the processes of engineering student data, building learning models, and discovering insights. This chapter describes the essential roles of each of the components. Later, the FIND framework is assessed on three different analytical models with three distinct student data representations. This framework was deducted throughout analysis and experiments and provides a foundational way to conduct learning analytics research.

FIND helps define the processes of learning analytics. The framework consists of three sub-components - 1) Student Data Engineering, 2) Learning Models Genera-

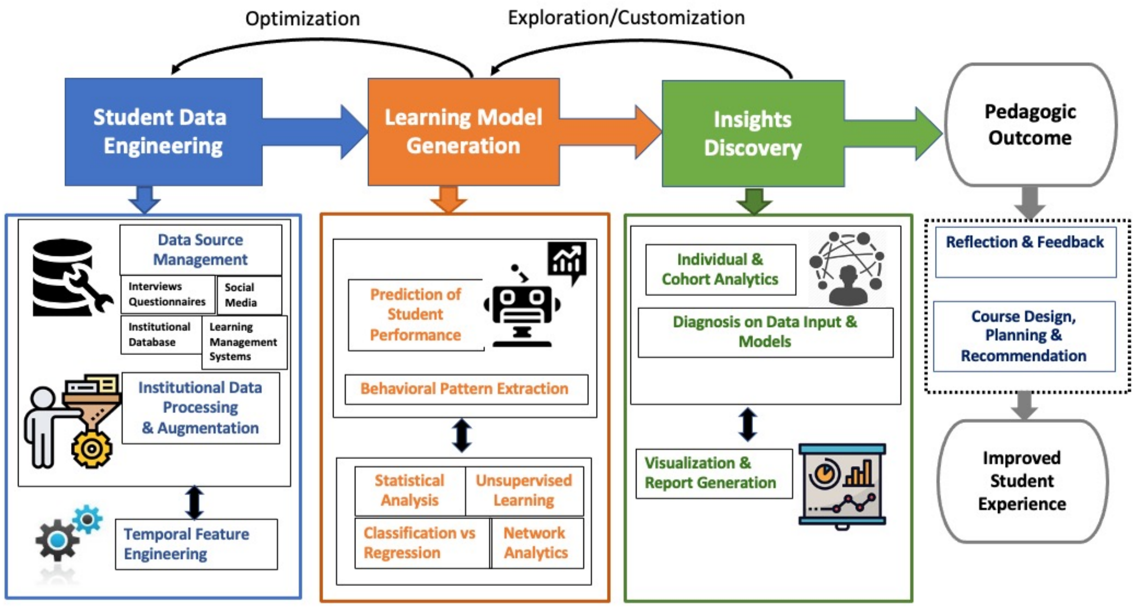


Figure 3.1: Our Framework for Learning Analytics. The framework comprises three main building blocks - 1) Student Data Engineering, 2) Learning Models Generation, and 3) Insights Discovery

tion, and 3) Insights Discovery leading to an optimal educational outcome. It not only represents a knowledge discovery cycle. The framework also emphasizes the cyclical nature of analytical processes and the ongoing need to refine and improve the system through successive phases of data engineering, processing, building models, and presenting/extracting information and insights through optimization and customization by exploration. Data engineering involves data selection, capture and processing, and aggregation of the data. Learning model generation includes the assembly and reporting of information and making predictions based on that information. Finally, insights involve the use, refinement, and sharing of knowledge to improve the system.

3.1.1 Student Data Engineering

This component encompasses all one needs to do, from extracting data from institutional and other databases to providing analytic models. It is one of the most challenging aspects. The type of data collected or preprocessed differs depending on the institution and the objective of existing applications. Nonetheless, it frequently includes information on performance, assessments, and activities from different timelines (semester-wise or weekly). Dealing with the big and sparse heterogeneous student data with class imbalance is a big challenge itself. Student data is not consistently populated for every student; data can be incomplete, inconsistent, entered incorrectly, and infrequently. Apart from sparsity and incompleteness, the data demonstrates heterogeneity due to several external reasons, including the frequent changes in program and curriculum, frequent course withdrawal, and change of majors.

Furthermore, student data is likely biased toward successful students, with much fewer students at risk, depending on how student risk is measured. Student risk is defined differently by different institutions. It may also differ within an institution, depending on the subject of study. Finding meaningful results from machine learning models and applications is nearly tricky without extensive data engineering and the discovery of appropriate data science models.

The FIND framework emphasizes the necessity of experimenting with various data models and representations. Two unique student data models are presented to demonstrate the importance of student data engineering. They are - 1) a network data model (see 4.2.1) [1] and 2) a temporal data model (see 4.2.2) [3].

A weighted student network is built from the student records for each semester recursively in the network data model. It reflects the connections between students. Six separate rules are used to identify common neighbors or affiliations in the student network: common courses, activities, advisers, department, majors, and pre-school information. After building the weighted student network, three different network features are retrieved besides the conventional academic data for the delayed graduation prediction. This strategy is unusual because it constructs the network without gaining entrance to evolving social media or interaction logs data from LMS.

The student temporal data model contains one temporal sequence per student(as illustrated in Figure 3.2), where a sequence represents nodes grouped in the sequential order of the enrolled semesters. Each node in a series represents a single point in time (i.e., a semester) and contains a vector of features, i.e., all of the students' data and activities. The student model mainly consists of three types of nodes:

1. The background node with all demographic, admission, and background information,
2. The semester node contains all activities and information for that semester, and
3. The result node contains all information on semester-by-semester aspects for performance evaluation, such as GPA and enrollment.

A student data model is provided that sorts heterogeneous data sources and creates information sequences for each student using time. The sequence data model can be used to assess the interdependence of events in a student's life. It allows analytics to uncover unexpected trends over time using student activity sequences.

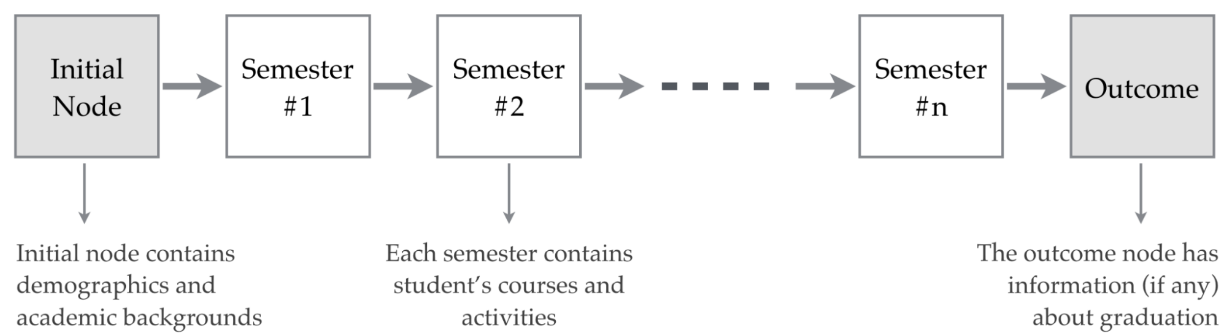


Figure 3.2: The Architecture of the Sequence Data Model [3]

Features are engineered that sustain temporal links in the student data for the unsupervised cluster analysis (see sections 4.2.2 and 5.2.1). Student data in semester nodes is used to generate all of the engineered features.

3.1.2 Learning Model Generation

The goal of this component is to create or find an analytic model for the student data. Statistical analysis, regression, classification, clustering, text mining, nearest neighbor analysis, network analysis, and model creation are some of the techniques used in learning analytics [23]. In this stage, researchers use one of the methodologies outlined above to evaluate performance or retrieve a meaningful behavioral pattern from the extracted characteristics using a statistical or machine learning method. A supervised machine learning algorithm, for example, can create a predictive model for classifying students based on course performance data. This trained and supervised model can then be applied to test data from unseen students to predict their performance earlier in their enrollment.

Another major problem in learning analytics research is determining the optimal model for achieving common goals with the best performance. To maximize outcomes, learning analytics (LA) analyzes preprocessed or engineered data to bridge relationships between learning activities and learning results. There are numerous ready-to-use machine-learning and deep-learning algorithms to handle predictive analytics and behavioral pattern extraction. With a substantial amount of labeled data, these strategies are utilized to address supervised classification or regression analysis problems. Due to the difficulty of evaluating unlabeled data, unsupervised machine learning and deep learning algorithms are less useful in production. Researchers sometimes use unsupervised clustering techniques to find complicated patterns in unlabeled data [14, 15]. As a result, using simply some arbitrarily labeled data, supervised learning might fulfill the prediction task. On the other hand, unsupervised learning may identify different non-trivial dependencies or even complicated patterns to get deeper insights [15].

Aside from the variety of applications of different analytical approaches, we've found that student data's heterogeneity and complicated structure add to the difficulty of choosing a suitable analytical model. Using explainable AI on a recurrent neural network model is another way to detect patterns and analytical operations for training and good predictive results. This dissertation uses XAI to compare the pattern of a single data point (local analysis or with local XAI) or a cohort of data points (global analysis or with global XAI) to the rest of the data. In other words, it detects if the learning model makes sense for an individual data point or a group of data compared to the rest of the data. The rationale behind this technique is the assumption that individuals in a cohort demonstrate the same behavioral patterns in the model's feature space, whereas each individual has some uniqueness.

3.1.3 Insights Discovery

This component focuses on how one can take advantage of learning analytics. Learning analytics technologies and methodologies can track significantly more data and underlying patterns than a single instructor, policymaker, or group of policymakers could. These tools and strategies can reveal previously unknown aspects of a student's learning experience and course or degree completion. Learning analytics is based on the notion that the data gathered through data engineering and analytics will aid in gaining insights into students' learning experiences and activities. LA analyzes the data to make predictions about students' performance and develop intervention strategies to improve pedagogical outcomes and learning experiences. Different analytical and pattern mining methodologies examine students' performance, estimated risks, behavior, and capabilities. Various performance measures are evaluated iteratively to improve the analytical method's performance.

Researchers use visual reports and visualization techniques to identify trends in data and their relationship to insights. They evaluate visual information and determine the potential for learning analysis cases to predict academic achievement [31]. For example, maintaining the size of the learning community and being able to do the additional processing associated with it is complex, especially distance learning can continue to be overlooked by most of the passive activities [31, 33, 32]. Visualization tools assist instructors and administrators in extracting and visualizing actor-specific and group-specific information exchange and performance analytics, which are typically difficult to observe at the abstract level.

Narrative results with explainable artificial intelligence are an alternative method for presenting complex and heterogeneous student data (XAI). The student data and analytic results are presented to decision-makers in this technique through insightful visualizations of the student performance pattern: for both individual and group settings. Explainable AI can also help data scientists in the training process of machine learning models in identifying if the model is improving over time. Explainable visualizations of the results provide a compelling, engaging, and easy way to understand complex and heterogeneous student data and the predicted results from a black-box model. Moreover, XAI makes it so effective in presenting complex student data in that it works for all decision-makers regardless of their level of expertise.

3.1.4 Pedagogic Outcome

The ultimate goal of the learning analytics approach is to devise interventions that would optimize the pleasant learning experience while also achieving pedagogical learning objectives. Giving informative feedback to suitable stakeholders like students or parents, course recommendations, course design, planning and scheduling, and suggesting new group formation to monitor changes in performance are all part of this process. Following thorough insight development, we strive to provide learners with automatic or manual feedback in response to their involvement in improving their performance and collaboration. Instructors redesign courses, organize, plan ahead of time based on the evaluation process results. They offer new groups and pedagogical activities to increase students' performance.

Chapter 4 to chapter 6 present the research and experimentation on degree-level analytics from different perspectives on three distinct learning models. The definition of a student's success or risk (the target variable for prediction) is presented in two ways: 1) by the number of years taken to graduate and 2) by using a Cumulative GPA threshold of 2.00. All of the models discussed preserve the temporal aspect of the features. All three approaches validate the three building blocks of the FIND framework - 1) by utilizing a novel model for student data representation discussed in Chapter 4, 2) by using innovative analytics models discussed in Chapter 5, and 3) by generating insights at the aggregate or individual student level discussed in both Chapter 5 and 6.

Chapters 4, 5 and 6 go over the components for each model, from data representation to generating insights in more detail.

CHAPTER 4: Student Data Representations

This dissertation evaluates the accuracy, interpretability, and prediction performance of several student data representations or models (current chapter: Chapter 4) and machine learning models (Chapter 5). For the various data representations and model techniques, the same data source is used in this dissertation.

In this chapter, different data models are developed for the different analytical approaches discussed in chapter 5. The data models change their representations depending on the input criteria of the analytical models, even though all of them are temporal, and they preserve the same information on the students.

4.1 The Dataset

For this dissertation research, the data mainly contains degree analytics-related features. The data consist of different course activities, semester-wise enrollment information, academic performance, and activities. These data are stored in a student administration database (Banner) and maintained by the school authority. For this research purpose and with the IRB approval, a data coordinator with access to the live database delivers the data as mentioned above on only undergrad students of the College of Computing Students of UNCC. The data includes the information from Fall 2004 to the current semester (Spring 2021) for this ongoing research. However, the analyses use the most recent 10 to 12 years of data to incorporate the upgraded course curriculum.

The dataset contains demographic, admission, and background information, semester-wise time-series information, and graduation information. Demographic and background data mainly consists of the data related to the ethnicity, gender, past and present geographic locations, marital and status of children, admission information while applying for UNCC, background information on the previous educational institutions. Semester-wise information is mainly temporal and cumulative over time for a single student. This information consists of the course activities, performance such as CGPA, individual GPAs of courses, academic standing, and all other information related to GPA type, instructor type, and course major. This dataset also includes graduation information such as expected graduation date and the date the student graduated, graduating CGPA, major, department, and academic advisors' information. We have access to the expected graduation date for ongoing students calculated from the required credits to graduate, which helps us determine a student's current state or progress.

4.2 Different Data Models

Three different data models or data representations are developed with the exact source of data. The data representations vary since the analytical models' output and analyses vary, and each expects a particular type of data representation. For the three different models, the following three distinct data models or representations are built-

1. Weighted Student Network Data Model [1]
2. Temporal Data Model with Engineered Features [5]
3. Mixed Data Model with Temporal and Non-temporal Features

4.2.1 Weighted Student Network Data Model

Section 5.1 presents a student network analysis-based prediction model for at-risk students who do not finish their degrees or graduate on time [1]. One of the leading hypotheses for this analytical model is that network features, or the students' connections increase early prediction accuracy on the potential dropout students when used with the conventional academic features available with the dataset. A cumulative (by time) weighted student network is built from the offline Banner database. It extracts the network features from the weighted network model. Each system in Banner, Advancement, Human Resources, Student, Financial Aid, and Finance has a view to an identity formation that allows for the viewing and updating of generic personal information [96]. For example, in Banner, the General Person Module is a single collection of data that includes a person's name, address, phone number, email address, and biographic information. A person's record is only produced once.

This analysis includes 13 years of data - from Fall 2004 to Fall 2016, including 2552 graduated undergraduate students. The study included only those students who have spent at least eight semesters in this school and have chosen a major from the Computing and Informatics College for at least one semester. On-time graduation is selected as a marker of success. The prediction algorithms are employed to identify students who took longer than six years to graduate or who could not finish their degrees in six years, including the dropped out or those who left the program after six years. Such at-risk students are approximately 30% of the total 2552 students. Students who have enrolled in the last six years are not included in the analysis. Therefore, the target variable for the supervised machine learning model is on-time graduation, in other words, graduation within six years.

The main contribution is forming a weighted student network model. Student networks are built based on student records rather than self-reported data from the learning management system (LMS) or social media logs. In addition to the conventional academic features, the network and fixed point features are extracted and used, which improve the prediction accuracy of the supervised machine learning model. The benefit of not using LMS data is that not all educators use the LMS similarly, resulting in inconsistent data for all students in a major. Moreover, using social media as a form of LMS or interactions has not become the norm yet. Therefore, the information on connections, collaborations, or interactions is not easy or consistent.

Demographics, pre-admission assessments, and institution information; academic features, for example, majors, courses attempted and completed, transferred courses, and advisers; and extracurricular projects, including intra- and interdisciplinary activities are collected.

In total, there are 57 features. The following are three kinds of features that were designed and extracted:

1. Academic features,
2. Student network features extracted from the weighted student network model, and
3. Fixpoint features

The first kind is already commonly used in learning analytics, but the second and third forms have only been employed in a few research projects. One of the critical contributions of this dissertation is feature engineering and academic activities based on student networks. The extraction of the latter two types of features is described in detail.

4.2.1.1 Academic Features

Several studies have used academic features such as grade-point average (GPA) to classify at-risk students [16, 97, 98, 99, 100, 71]. It includes students' demographic and personal information (such as age, citizenship, gender, primary ethnicity), high-school or previous institutional records (school rank, percentile), and academic progress over semesters (such as course-wise and semester-wise GPA, attempted and passed credits, number of courses taken per semester, information on transferred, withdrawn and failed courses).

4.2.1.2 Student Network Features

Each semester, a weighted student network is created. The *cumulative intensity* of the edge (or link) between two students is represented by the edge weight value between them at the semester t. Their intensity will rise over time because it is cumulative. The edge weight between two students x and y at semester t, abbreviated as $w(x, y, t)$, is determined as follows:

$$w(x, y, t) = \exp\Big(\sum_i rescale(normalize(w_i(x, y, t)))\Big),$$

$$\begin{aligned}
\mathrm{w}_1(x, y, t) &= \textstyle\sum_{t' \leq t} \frac{C(x,t') \cap C(y,t')}{C(x,t') \cup C(y,t')}, \\
\mathrm{w}_2(x, y, t) &= \textstyle\sum_{t' \leq t} same_activity(x, y, t'), \\
\mathrm{w}_3(x, y, t) &= \textstyle\sum_{t' \leq t} same_advisor(x, y, t'), \\
\mathrm{w}_4(x, y, t) &= same_dept(x, y), \\
\mathrm{w}_5(x, y, t) &= same_major(x, y), \\
\mathrm{w}_6(x, y, t) &= same_age_high_school(x, y),
\end{aligned} \tag{4.1}$$

$C(x, t)$ signifies a set of courses taken by student x during semester t in equation 4.1. The indicator function $w_i(x, y, t)$ will return one if x and y are two different students with the same i) activity, ii) adviser, iii) department, iv) major, or v) high school. If they do not satisfy these six rules, the weight(w_i) will be zero, meaning they are not connected or are not neighbors. The high school records are non-temporal information. That is why they are not cumulative (only used once) while calculating the weights by this rule. All the other five rules are dependent on time. For example, the $w_1(x, y, t)$ inspired by the Jaccard index [101] calculates the total of the common course ratio between x and y till t. After normalization, w_i ranges from

$[0, 1]$, and the mean of w_i at t becomes 0.5 after rescaling. Jaccard index[1] is a similarity metric for nodes in a network that defines the number of shared neighbors divided by the sum of all neighbors. Some weights have bigger scales than others, and without normalization and rescaling, they may dominate the final weight value. Edge weights are standardized to prevent this. As a result, each wi becomes equally essential in our defining the edge weights. The term "activity" refers to most of the college's student-focused extracurricular organizations, athletics, and programs. The exponential function strengthens the significant connections. As a result, a student network with t edges may have a large number of small-weighted edges and a small number of large-weighted edges.

For each semester t, a student network is created, with over 40 networks generated from 13 years of student records. Figure 4.2 depicts a student network with three different edge colors: orange for successful students, blue for at-risk students, and dark green for at-risk students for several classes. It's worth mentioning that many students at risk lack strong ties with their successful peers.

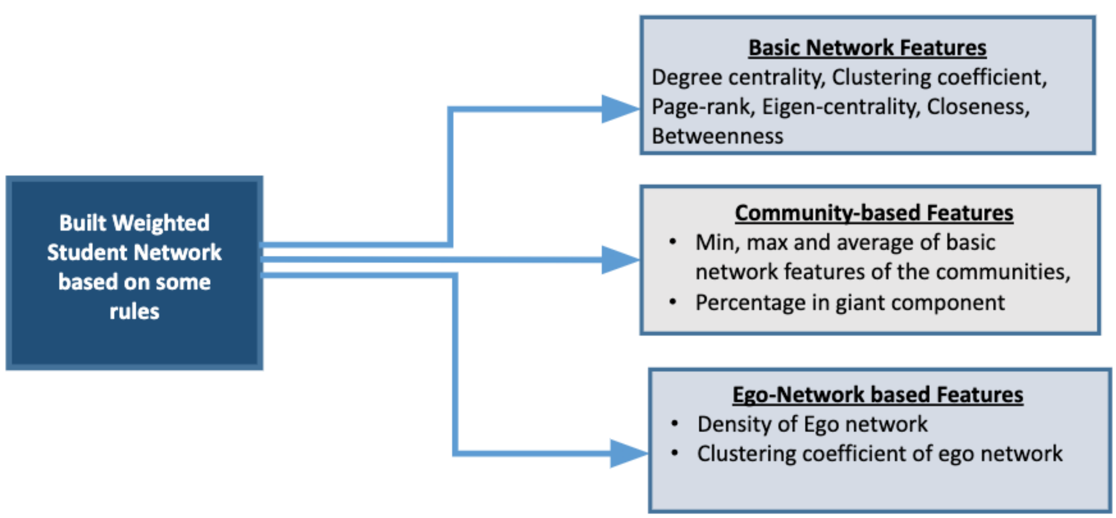


Figure 4.1: Extracted Network Features extracted from Built Student Network based on Equation 4.1

[1]The Jaccard index is a popular node similarity metric in networks based on the number of common neighbors divided by the sum of all neighbors.

After successfully building the weighted student network from the student records, we then extract three types of (See figure 4.1) network features from that network as follows :

1. Basic Network Features,

2. Community-based Network Features,and

3. Ego-network Based Features

I) Basic Network Features: The term "centrality" under network analysis refers to approaches for determining the relative importance of nodes (i.e., students are the nodes in our case).

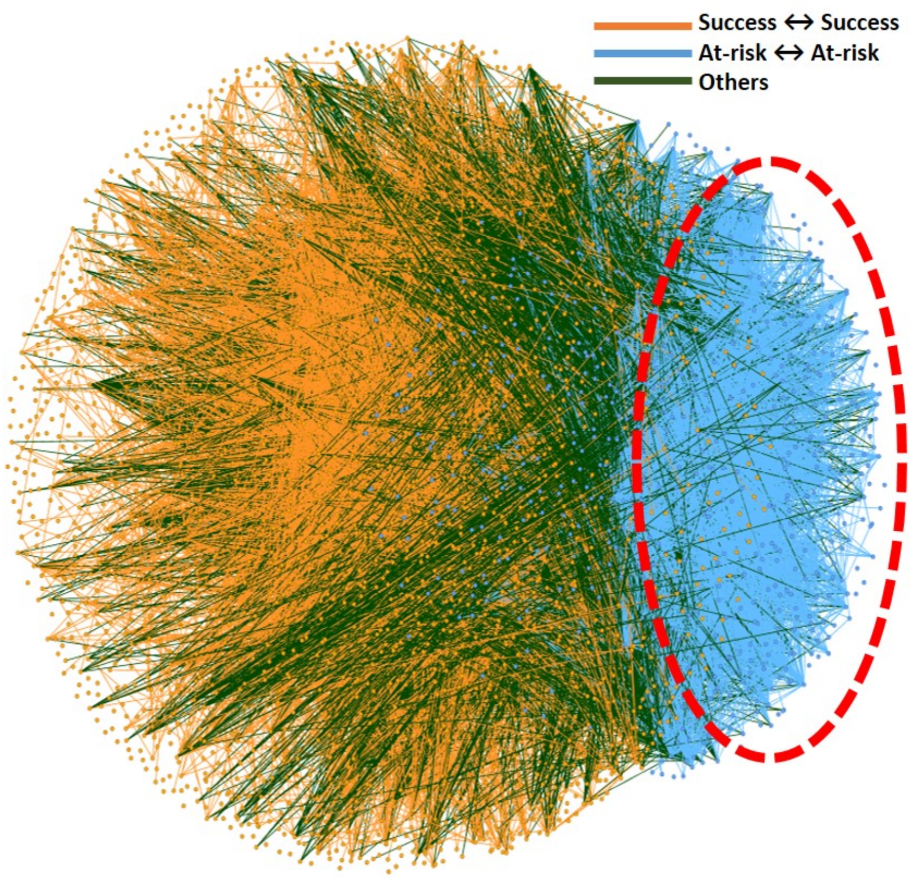


Figure 4.2: The student network is edge-colored after filtering out weak (90th percentile edge weight) connections. Within the dotted red area, many at-risk students lack deep ties with successful classmates [1]

Degree centrality [102], closeness centrality [103, 104], betweenness centrality [105, 106], and PageRank [107] are all examples of centrality measurements. The centrality measurements are chosen after statistical evaluations such as the t-test and histogram, which provide sufficient and noticeable information to distinguish at-risk students. Degree centrality, or the amount of links occurring upon a node, is one of the most basic characteristics. The number of times a node functions as a bridge between other nodes is determined by its betweenness centrality. Practically every concept of centrality has an unweighted version and a weighted version. All centrality indicators adopted in this study are weighted unless otherwise noted. Figure 4.3 demonstrates some basic network features in a network.

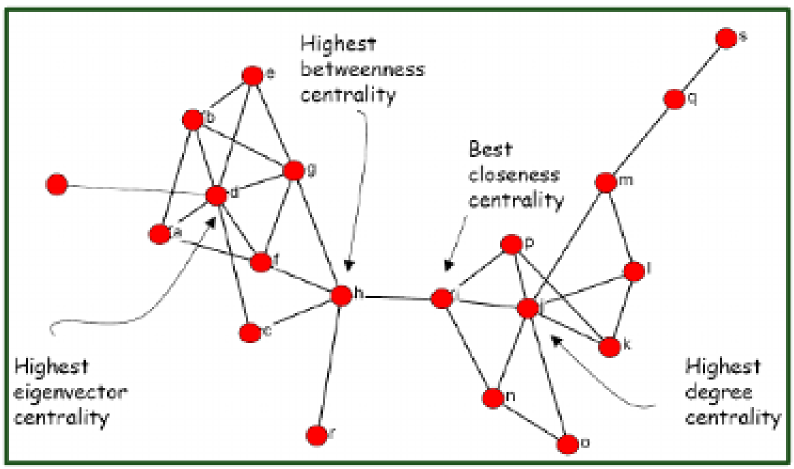


Figure 4.3: Different Centrality Measures (basic network features) [4]

II) Community-based Features: In network analysis, community detection is a well-established research area. It is occasionally used as a subroutine to solve other issues, such as the one we're working on [108, 109, 110, 111]. Because a student can join many overlapping communities, we use overlapping community detection methods. For example, a student A can share connections with two different students, B and C, who took two courses separately with A. Because of its accuracy and popularity, we chose SLPA [112] as our base community detection approach. We calculate the following features after identifying several overlapping communities in each network at the time t (semester), N_t:

1. Let $Com(x) = \{Com_1, Com_2, \cdots\}$ - a collection of communities in which x is an active student. Finally, min/max/average over the $Com(x)$ communities are aggregated for each variety of network feature (x)

2. A giant component in each network at the time t (semester t), N_t denotes the most significant and most prominent community. In many circumstances, a student is checked if they are a member of that component because it is one of the most visible student groups. The proportion of such cases is then calculated over a period. Students with a ratio of 1 are stable members of a giant component throughout their academic background.

III) Ego-Network-based Features: An ego network is built for each node for each time point, meaning each student will have an ego network at each semester. Ego network represents one node's induced sub-network surrounded by its only neighbors [113]. The ego-network density (see equation 4.2) and clustering coefficient from each node's (students') ego network are extracted as ego network features [103, 104].

For any weighted network (N) at time t, density is defined as follows:

$$Density(x,t) = \frac{2\sum_{(a,b)\in Ego(x,t)} w(a,b,t)}{|Nei(x,t)| \cdot (|Nei(x,t)| - 1)}, \tag{4.2}$$

In equation 4.2, $Ego(.,t)$ refers to a set of one's neighbors in N_t, whereas $Nei(.,t)$ refers to the edge set of one's ego-network in N_t. Recall that the final weight in Equation 4.1 is cumulative and uses the exponential function. As a result, if long-term connections for course registration and student activity are not maintained, ego-network density will have a low value. Therefore, it is speculated from the outset that successful students may have thick ego networks.

4.2.1.3 Fix-point Features

A fixed point is a constant feature where the input and output of the stationary system are the same [114]. For example, a fix-point x signifies $x = f(x)$ when given a function $f(.)$.

Fix-point calculations are employed in a variety of fields. The stationary distribution of the Markov chain, i.e., $\pi = \pi\mathbf{P}$, where $\mathbf{P}$ is the transition matrix, is a good example. The authors created a mutually recursive complex system of variables [107, 115, 56, 116, 117, 118, 119], and their fix-point values are used to interpret vertices. Domain-dependent knowledge is required to define such a complicated and stationary variable system. The educational experiences and access to the available data provide the domain knowledge. Here, the domain knowledge is based on two fundamental assumptions:

1. Many students enrolled in the same courses, activities, or undergraduate advisers have similar features or common characteristics.

2. Courses, activities, or advisors that students have taken might be used to characterize their traits [120, 25]. The network features consider the interactions between students. However, without considering other students, the fix-point features identify students based on course, activity, or advisor data. Hence, building a network for extracting fix point features is unnecessary.

$$\begin{aligned}
\mathrm{val}(\mathrm{c}_i, t) &= \sum\nolimits_{c_j} \frac{\#stu(c_i, c_j, t)}{\sum_{c_k} \#stu(c_k, c_j, t)} val(c_j, t), \\
val(a_i, t) &= \sum\nolimits_{a_j} \frac{\#stu(a_i, a_j, t)}{\sum_{a_k} \#stu(a_k, a_j, t)} val(a_j, t), \\
val(v_i, t) &= \sum\nolimits_{v_j} \frac{\#stu(v_i, v_j, t)}{\sum_{v_k} \#stu(v_k, v_j, t)} val(v_j, t),
\end{aligned}
\qquad
\begin{aligned}
\mathrm{val}(\mathrm{s}_i, t) &= \sum\nolimits_{c_j} take(c_j, s_i, t) \frac{1}{\#stu(c_j, t)} val(c_j, t) \\
&+ \sum\nolimits_{a_j} take(a_j, s_i, t) \frac{1}{\#stu(a_j, t)} val(a_j, t) \\
&+ \sum\nolimits_{ad_j} take(v_j, s_i, t) \frac{1}{\#stu(v_j, t)} val(v_j, t),
\end{aligned}$$

(4.3)

Input: Student network $N_t = (V, E)$, Course and Activity Records
Output: $val(s_i, t)$ for each student

Initialize $val(x, t) = \frac{1}{n}$ where n is the total number of courses, activities, advisers, or students depending to the type of x.
while *until the convergence of* $val(s_i, t)$ **do**
  Update $val(c_i, t)$; Update $val(a_i, t)$; Update $val(v_i, t)$; Update $val(s_i, t)$
return $val(s_i, t)$

Algorithm 1: Algorithm for Calculating Fix-points in our Complex Variable System [1]

It is worth noting that our variable definitions closely reflect our domain knowledge. Because of the normalization coefficient, if two courses, activities, or advisers have many students in common, their values will be pretty comparable. The sum of all course/activity/advisor values determines a student's value, which is purely based on the student's courses/activities/advisors. Numerous variables are defined that are mutually recursive based on these intuitions. This variable's value is shared with nearby variables as equation 4.3.

The letters c_i, a_i, v_i, and s_i stand for course, activity, adviser, and student, respectively. $students(x, y, t)$ returns the number of students who enrolled in two courses, activities, or advisers x and y during semester t, where $take(x, s, t) \in \{0, 1\}$ indicates if students enrolled in course, activity, or adviser x during semester t. As a result, $val(x, t)$ reflects the influence value possessed by each entity x during semester t. Since we establish a network N_t each semester, these variables are defined for each semester. Department, major, or degree information is not used in the variable definition because they are too broad.

The fixpoints of the influence variables are calculated using Algorithm 1. The variables are updated using an iterative process. The convergence of the student variables is essential. Additional features are the converged student values. Its convergence is ensured by the following theorem (Theorem 1), and there is only one solution.

Theorem 1. *The proposed complex variable system has one unique convergence point.*

Proof. Each course, activity, or advisor variable type has its own Markov chain, with

Table 4.1: Feature Vector for Predictive Analytics

Type	Student ID	Semester Number	Academic Features	Network Features	Label
Train	1	1	GPA, Success Ratio, etc.	Degree, Density, etc.	S
Train	1	:	:	:	S
Train	1	8	GPA, Success Ratio, etc.	Degree, Density, etc.	S
Train	2	1	GPA, Success Ratio, etc.	Degree, Density, etc.	R
Train	2	:	:	:	:
Train	2	20	GPA, Success Ratio, etc.	Degree, Density, etc.	R
Train	:	:	:	:	:
Test	2000	1	GPA, Success Ratio, etc.	Degree, Density, etc.	S
Test	2000	2	GPA, Success Ratio, etc.	Degree, Density, etc.	S
Test	:	:	:	:	:

each variable being a state of the Markov chain. The coefficients of the variables in Equation 4.3 define the transition probability in each Markov chain between two states. All of the coefficients are normalized, and the sum of their outward transition probabilities is 1.0. As a result, the entire procedure is the same as calculating the stationary distribution in a Markov chain.

It is commonly known that if all states are strongly coupled in a finite number of transitions, a Markov chain has only one unique stationary distribution. Student values are simply the total of other courses/activities/advisors variables, and they will converge if the others do. Each semester, the data was evaluated to see if it corresponded to the strongly connected Markov chain situation, which it did due to the high population and density. As a result, each semester, the data instance has a single fixpoint.

□

After extracting the three types of features, a feature vector is formed with them (See Table 4.1), where each row represents a studentâs data for a single semester. It indicates the input data format for the supervised machine learning models on the network student data representation (See section 5.1) for predicting on-time or delayed graduation.

4.2.2 Temporal Data Model

This section presents the data model used for both the unsupervised K- means cluster analysis (See section 5.2.1) and the Long Short Term Memory (LSTM) recurrent neural network model (See section 5.2.2). A temporal-sequential data model built by another former Ph.D. student influences this data representation (See figure 4.4) analysis [3]. One sequence per student is restrained in the student temporal/sequence data model, where a sequence represents nodes arranged in the sequential order of the enrolled semesters. Each node in a series represents a single point in time, e.g., a single semester. It contains a vector of features, such as all of the students' data and behaviors. There are three sorts of nodes in the student sequence data model: 1) the background node with all demographic, admission, previous education, and background information, 2) the semester node with all semester-wise(mainly the time-series data) activities and information (such as individual course performance data, semester-wise CGPAs, academic standings), and 3) the outcome node with all information on the semester-wise features for performance evaluation such as GPA, number of years of enrollment.

This hybrid data model obtains three types of nodes, and the features are tweaked with upgraded information or more features are added to the model. Temporal engineered features are calculated from them too. The unsupervised cluster analysis and the RNN model both utilize these engineered temporal features.

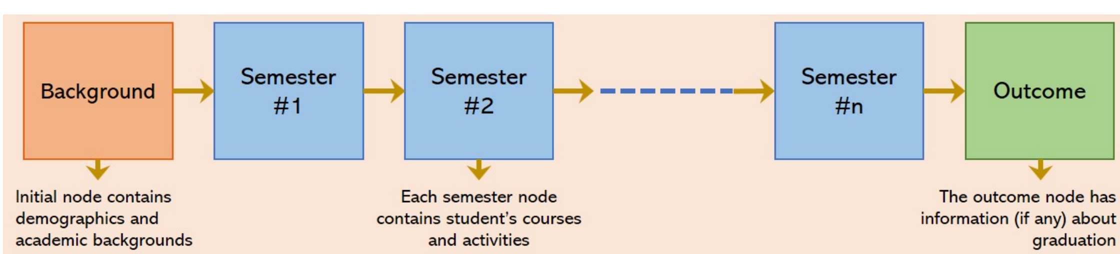


Figure 4.4: Student Sequence Data Model [3])

4.2.2.1 Types of Features in Student Sequence Data Model

There are three sets of available features in the student sequence data model. Features for each node will be chosen from those sets. First, a student data model is offered that uses time (in our case, time=semester) to filter heterogeneous sources of student data and build information sequences for each student. It allows analytics to examine the interconnection of events in a student's life and identify unanticipated patterns based on temporal sequences of student behavior using the sequence data model. Sorting student behavior through time and aggregating the data in a sequence style helps account for time dependency in analytics and build data nodes, contextual information within nodes, improve granularity, and interpret sequences as narratives. Options are provided to select the features to include in the analytics on the landing page of our interactive application. The student model is made up of three types of nodes: the non-temporal background node, which contains all demographic data, the semester node, which includes all semester-specific activities and data; and the outcome node, which consists of a mix of temporal (i.e., CGPA for each semester) and non-temporal (i.e., number of years to graduate) features.

I) Non- temporal Background Features The background features consist of demographic, admission, and background information of the students. Demographic and background information consists of gender, ethnicity, past and present demographic locations, marital status, number of children, etc. Admission information consists of preschool or previous education information such as the location of the former education, various performance metrics such as CGPA, percentile, and school rank. The background information of multiple schools and colleges is from different times. Since this information does not change for a student and is recorded just once, these features are considered constant and non-temporal features.

II) Semester-wise Temporal Engineered and Raw Features The semester features consist of semester-wise information such as all the course information, cumulative GPA (CGPA), semester-wise class rank, major, progress evaluation, academic standing, etc. It also includes course performance features such as individual GPAs earned for courses, credits attempted, credits earned, and GPA types.

Features are engineered that maintain temporal relationships in the student data for the unsupervised cluster analysis. All of the engineered features are derived from student data in semester nodes. A 2D feature vector is formed for each feature. It has temporal values for each student's engineered features for their first eight semesters (See figure 4.5). The feature vectors are generated for those students who have been enrolled for at least eight semesters. The summer semesters are not considered in the dataset since the summer semesters represent an inconsistent and sparse dataset. Moreover, students generally retake the courses in the summer semesters, which they failed or withdrew previously.

CGPA (Cumulative Grade Point Average) is considered as the target variable for the latter two analytical models. K-means clustering analysis is an unsupervised machine learning approach. However, the cohorts of students are annotated with a CGPA threshold of 2.00 in their eighth semester. If the students have a CGPA greater than 2.00, they are considered as successful. Otherwise, they are considered as students at risk. The analysis needs to find how the two cohorts (successful and at-risks) form the clusters with a particular engineered feature.

One of the engineered elements that produce significant student clusters for the performance criteria (e.g., CGPA of threshold 2.00) is "Course Progression Through Semesters." This feature displays a normalized value for the sum of all courses taken in a semester divided by the course level (for example, the course level for ITSC 1200 is 1). The fundamental hypothesis is that this attribute should be either constant or rising for successful students. If the feature declines from semester to semester, the student is either not advancing to higher-level courses or is repeating or retaking some lower-level courses. The other temporal engineered features are listed in the following section (Section 4.2.2.2).

Some raw numeric temporal data are employed for the recurrent network analytics described in section 5.2.2. Some of the engineered features from the unsupervised learning model are reused as input to the RNN model. Some of the textual or categorical features such as âacademic standingâ are converted to numerical values as categories (with One hot encoding) since the neural network model cannot work with textual data.

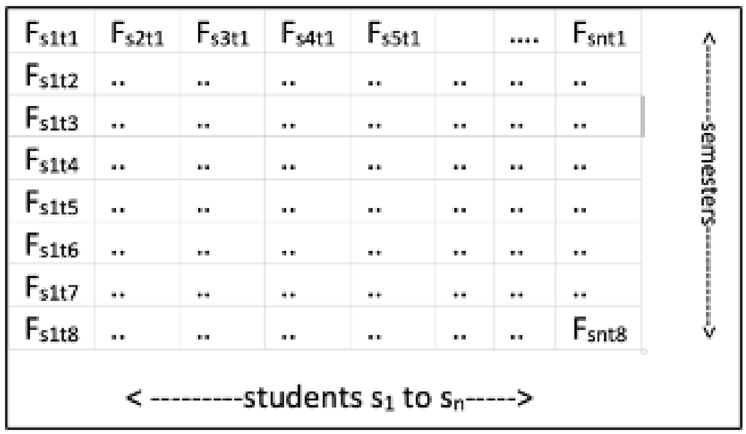


Figure 4.5: Input to K-Means Clustering : A feature vector representation for feature F denoting 8 semesters($t1tot8$) of its values vertically and for n number of students horizontally [5]

III) Hybrid - Temporal and Non-temporal Outcome Features Outcome features consist of the semester-wise features for performance evaluation such as GPA, the number of years of enrollment, expected graduation date, total credits attempted, and total credits passed. It is a hybrid of constant and temporal features. For example, each student has GPAs for each semester which is temporal, but the total number of years a student takes to graduate is fixed and does not change over time.

4.2.2.2 Datasets for Student Temporal Model

The same set of features is collected from the same data source for a varying amount of time. The time lengths vary for the two different learning models as mentioned above.

Dataset for Unsupervised Clustering Analysis: The unsupervised K-means clustering model[5] includes 6203 students. The data is retrieved from the same Banner data source (See section 4.1). Data throughout ten and a half years are collected- from Fall 2008 to Fall 2018. The temporal and semester-wise features are solely used from the data to construct a feature vector for each student in terms of the engineered temporal features listed below for the unsupervised learning model:

1. Course level progression through semesters
2. Academic standing progression through semesters
3. Percentage of transferred courses to the total number of courses per semester
4. Percentage of courses with an A grade to the total number of courses per semester
5. Percentage of courses with a B grade to the total number of courses per semester
6. Percentage of failed courses to the total number of courses per semester
7. Percentage of course withdrawals to the total number of courses per semester

For each student's first eight enrolled semesters, the feature vector contains the values for one of these engineered features.

Dataset for Supervised Deep Learning Model (RNN): A supervised classification model with a vanilla LSTM RNN is developed (see section 5.2.2.1) for this dataset. The dataset includes 6220 students. In addition to the features used in the unsupervised clustering, some other categorical and raw temporal features are used for this analysis. Some non-temporal features were used at first, which resulted in model overfitting and less explainability. Moreover, non-temporal features are irrelevant and not suitable for temporal models like LSTM RNN.

Due to the limitations and observations found, more data may help this model not to overfit and have interpretable results. The analysis is changed to a regression model with more feature engineering and hyper-parameter tuning. Data from three more semesters are added, which results in 6491 students in total for the bidirectional LSTM analytical model.

The features set used in the RNN analysis is provided in appendix A.1. Besides these, features deducted by sparsity and feature engineering are also listed.

CHAPTER 5: Machine Learning Models for Student Success Prediction

This chapter presents three machine learning approaches to predict success and risk and analyzes student data models presented in chapter 4. This chapter focuses on degree-level analytics. The definition of a student's success or risk (the target variable for prediction) is presented in two ways: 1) by the number of years taken to graduate (i.e., six years) and 2) by using a threshold of cumulative grade point average (CGPA) of 2.00. All of the three models discussed in this chapter preserve the temporary aspect of the features.

5.1 Network Analytics: Predict On-time Graduation using Temporal and Weighted Student Network Data Model

To predict the risk of delayed graduation earlier in the academic career and investigate the patterns of connections and collaborations among those at-risk students, we extract a feature vector of three different features discussed in section 4.2.1 [1]: 1) Weighted student network features, 2) conventional academic features, and 3) fix-point features that are extracted from the temporal and cumulative student network data model.

Only 60% of undergraduate students in the United States complete their bachelor's degrees in six years [121]. The rest take more than six years to complete their education or never complete it at all, eventually dropping out. A prediction model based on student network analysis is provided for detecting delayed graduation instances [1]. An innovative network analytic methodology is presented for predicting students at risk who might drop out of college due to taking longer to graduate (more than six years). Moreover, the early analysis finds out which students are prone to drop out eventually, either early or later, after six years.

The main contributions of our network analytics approach are -

1. The network analytics model is unique and different from other network analytics approaches. Diverse data from students' collaboration and activity records are leveraged to determine student network features besides the typically utilized academic features such as GPA and credits earned, (inconsistent) social media, and LMS log data.

2. Starting from the fifth semester (early prediction), we identify at-risk students who might take longer than six years to graduate or who are likely to drop out sooner or later.

3. Fix-point feature, a special type of feature, is generated in this research (see Figure 5.1).

4. This analytical approach achieves a better performance than using only conventional academic features. It accomplishes an F-1 score of 0.85 and an AUCROC of 0.86 by combining the network and fixed point features with the academic features.

5. The findings include compelling features and risky behavioral patterns in the cohort of the at-risk students, which will help the domain experts make informed decisions on guiding the students. At-risk students, for example, have a large number of weak connections rather than a small number of strong ones.

Network features such as degree centrality and the ego-network density of at-risk students differ significantly from successful students.

This section will discuss the overall workflow of our analysis, the prediction results, and the statistical analysis of the weighted network. Throughout the discussion, we answer some research questions to demonstrate the efficacy of our approach.

5.1.1 Overall Workflow of the Network Analysis Framework

The whole workflow of the weighted student network analysis is shown in Figure 5.1, from data production through prediction results generation.

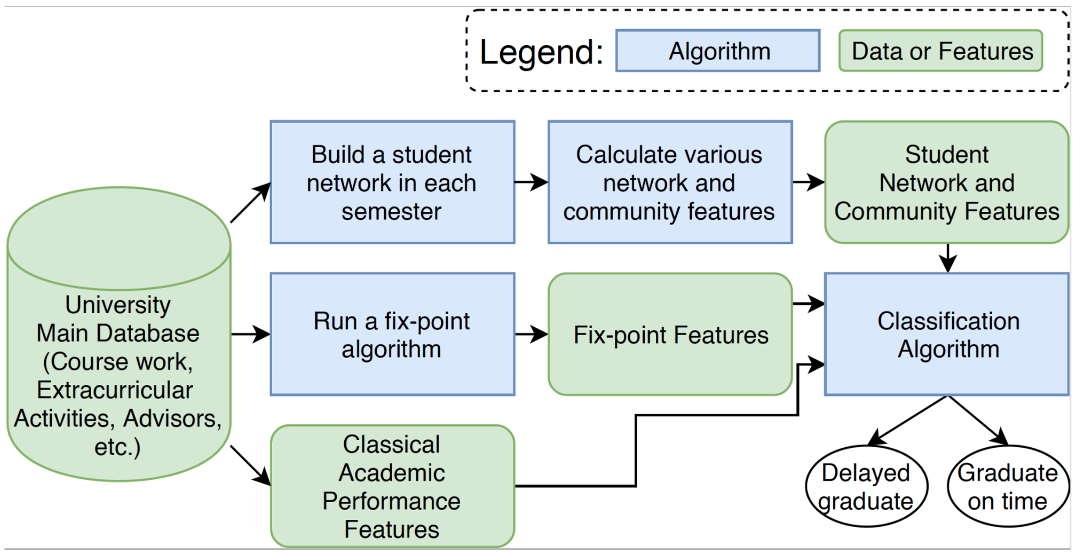


Figure 5.1: The Overall Workflow of the Weighted Student Network Analysis Framework [1]

Chapter 4.2.1 already covered how to create the network data model from student records and extract three types of features: academic features, student network features, and fix-point features. The data we collected for this analysis is from a longer time frame, i.e. over 12 years. Several curriculum changes took place within this timeframe. The training set includes the students who graduated before Spring 2013, and the test set includes everyone else. The train's to test ratio is 77:23. Grid search with 10-fold cross-validation is used to determine the best model. SVM, Random Forest, Decision Tree, AdaBoost, RBM, Bagging, Multi-Layer Perceptron, and more classifiers are all put to test. Two classes in the training set are notably imbalanced, with 63 percent successful and 37 percent at-risk. Under/oversampling techniques are employed to solve the imbalance issue[122]. Random Forest succeeds effectively, as shown by the fact that it generates all of the values shown in Table 5.1. This study distinguishes between successful students who completed their education in less than six years and at-risk students who completed their education in more than six years.

5.1.2 Research Questions

The following research questions are answered throughout the predictive and network analysis:

- **RQ-NA1:** Does including students' network features aid in the identification of at-risk students?
- **RQ-NA2:** Is it possible to identify at-risk students earlier by using their social network features?
- **RQ-NA3:** To what extent do successful students' network connections differ from those of at-risk over time?
- **RQ-NA4:** Who are the active members of student communities, and who are the peripheral members? How do successful and at-risk students interact in their communities?

5.1.3 Analysis on Prediction Results

The network features are calculated in a variety of student networks whose edge weights are determined using either a single rule (i.e., six rules for connecting edges: shared courses, advisors, activities, department, major, and high school information) or the combined weight (as indicated in Table 5.1). Prediction results for two different periods are presented: 1) for their overall academic career from admission to graduation and 2) early prediction, only until their fifth semester of data. Table 5.1)does not include three rules for producing edge weights (such as common department, major, and high school information) since they are sparse and/or perform poorly.

The best performance among them is shown by network features based on the aggregate weight. The recalls are 56% and 50% of at-risk students across all and early phase students, respectively, using the academic features only. The addition of

Table 5.1: Prediction Results (*The definitions of weights are provided in section 4.2.1, where we talked about the building process of network model) [1]

	F-1 Overall	AUCROC	Recall of At-risk	F-1 of Successful
	For Overall Academic Career(Admission to Graduation)			
Academic Features	0.78	0.76	0.56	0.78
Network Features (course only)	0.76	0.72	0.62	0.83
Network Features (activities only)	0.73	0.66	0.51	0.81
Network Features (advisors only)	0.74	0.70	0.61	0.81
Network Features (all weights considered)	0.81	0.8	0.64	0.87
Academic + Network Features	0.84	0.86	0.69	0.89
Academic + Network + Fix point Features	**0.85**	**0.86**	**0.70**	**0.89**
	Early Phase (After 5th Semester)			
Academic Features	0.8	0.75	0.5	0.8
Network Features (course only)	0.8	0.71	0.54	0.86
Network Features (activities only)	0.75	0.66	0.51	0.82
Network Features (advisors only)	0.76	0.68	0.55	0.83
Network Features (all weights considered)	0.8	0.77	0.55	0.79
Academic + Network Features	0.84	0.85	0.56	0.9
Academic + Network + Fix point Features	**0.85**	**0.85**	**0.58**	**0.9**

the network features obtains 69% and 56% recalls for the at-risk student class and 70% and 58% recall after applying all accessible features. On additional criteria like $F-1$ and $AUCROC$, the prediction model, which contains all academic, network, and fix-point characteristics, outperform others significantly. These findings strongly qualify the solutions to our study questions **RQ-NA1** and **RQ-NA2**. Therefore, network features help with overall prediction and, more specifically, forecast across shorter periods.

5.1.4 Statistical Analysis on Weighted Student Network

Statistical t-test analysis on both successful and at-risk students is conducted for each network feature and the fix-point features. Three types of network features are extracted:

1. Basic network features or centrality measures,
2. Community-based centrality measures obtained by the overlapping community detection algorithm SLPA, and
3. Ego-network-based features

The findings, in terms of each of these features, are listed here.

1. Basic Network Features

The total of the edge weights related to a student's neighbors at the time "t" (or a semester) is their degree centrality. Previously, the assumption was successful students have more connections and thus higher degree centrality scores.

Findings, on the other hand, refute the notion. Degrees from several perspectives are shown in Table 5.2. Average values are calculated for the best 50% and most deficient 50% of students in each prediction class in terms of mean degree (successful vs. at-risks). Their mean values are substantially different, with successful students scoring **2522.4** and at-risk students scoring **5258.1** (p-value < 0.01). Some students who are at risk have more interactions than the majority of their peers.

The analysis can be explained in a variety of ways-

- For example, some at-risk students participate in an excessive amount of extracurricular activities. Similarly, the bottom half of at-risk students had significantly fewer degrees than the bottom half of successful students, **557.4** vs. **373.1** (p-value 0.01). It also implies that many at-risk students have fewer opportunities to interact with their peers.

- Although the significance level of the successful student is low (p-value > 0.01), they had higher mean values than students at risk in the top 50% and bottom 50% cases of early phase students. However, the average degree of successful students in the fifth semester is greater than the overall mean degree (**3106.5** vs. **25222.4**). It's only conceivable if successful students establish strong bonds early in their academic careers and ii) don't build many new connections later. On the other hand, at-risk students must repeat some courses to pass them, resulting in many links in our network because they repeat the same courses more frequently than successful students.

- The mean degree of the top 50% of students at risk is **5392.7** in the fifth semester, but it rises to **10143** when their academic career is considered. It demonstrates that they interact with many new students for student activities, courses, and advisers even in their late academic periods. They encounter new people (classmates) since they retake classes frequently, change advisers, and their previous successful classmates have already progressed in their academic careers.

- Surprisingly, the average degree of the most deficient 50% of the low performing students did not vary significantly over time, ranging from **311.7** after the sixth semester to **373.1** over the academic year. They are cut off from the rest of the world regularly.

These findings confirm our research question **RQ-NA3** in that successful and at-risk students behave differently over time. We also believe that successful students have higher values in terms of other network features.

However, a few of our findings contradict one another, indicating the variety of behaviors among at-risk students. For example, at-risk students can serve as community connectors. For example, the top half of at-risk students in the prior degree centrality study continually create new connections (rather than staying in a community). Therefore, they will have a greater PageRank, betweenness, and eigenvector centrality than successful students. No significant changes are found in the population's bottom half.

Closeness centrality [123], clustering coefficient [124], leverage centrality [125], and other measures of centrality are also considered. We don't see any significant differences between the two student cohorts for some of those centrality features.

2. Community-based Centrality Measures

Table 5.2 shows that successful students have higher mean values for community features, indicating that they play a more critical role in their communities. It also corresponds to the high betweenness centrality scores for the low-performing students,

Table 5.2: Statistical t-test results for mean centrality, mean community-based features, and two student classes. Except for the situations noted with boldface writing, all p-values are less than 0.01[1]

		Successful (entire period)	At-risk (entire period)	Successful (at 5th sem.)	At-risk (at 5th sem.)
Degree	All	2522.4	**5258.1**	**3106.5**	2856.6
	Top 50%	4487.3	**10143.0**	**5575.5**	5392.7
	Bottom 50%	**557.4**	373.1	**634.8**	311.7
Page Rank	All	0.00075	**0.00202**	0.00072	**0.00193**
	Top 50%	0.00118	**0.00365**	0.0011	**0.00346**
	Bottom 50%	0.00031	**0.00038**	0.00032	**0.00038**
Eigen.	All	0.0068	**0.02212**	0.0058	**0.02323**
Betw.	All	0.00079	**0.00456**	0.0006	**0.00391**
Close.	All	**0.4357**	0.4281	**0.4372**	0.4214
Min Degree	All	**553.3**	498.8	**568.8**	423.1
Min Eigen.	All	**0.5348**	0.5145	**0.5419**	0.4719
Giant.	All	0.0674	**0.087**	0.0665	**0.0929**
	Top 50%	0.1268	**0.1606**	0.122	**0.1713**
	Bottom 50%	0.0074	**0.0131**	0.0077	**0.0137**
Fix Point	All	0.000601	**0.000722**	0.001303	**0.001793**
	Top 50%	0.00115	**0.00178**	0.002344	**0.00541**
	Bottom 50%	**2.144450e-81**	1.561194e-81	**2.511162e-131**	1.818025e-131

implying that they are joints between groups rather than center members. Furthermore, they are more likely to be members of the larger communities than successful students since they engage with numerous groups and thus end up in peripheral positions in communities. Surprisingly, the average degree centrality of the bottom 50% of at-risk students is lower than the bottom 50% of successful students. Despite this, they have a higher average proportion of participation in big communities. At-risk students may go to many places but build few connections within them.

3. Ego-network-based Feature

The clustering coefficient and the ego-network density are mutually beneficial. Ego-network density can be high if some edge weights are large. However, some high edge weights do not always result in a clustering coefficient of one. Only when an ego network is complete does its clustering coefficient become one. Because of this trait, larger values of the clustering coefficients are expected from successful students than from at-risk students. Successful students' "Ego Networks" are "more comprehensive" than those of at-risk students. If long-term connections in numerous activities are not maintained, ego-network density will have lower values. As a result, we hypothesize that successful students have large ego networks. In all cases, successful students had a higher ego network density than at-risk students (**3.634** vs. **3.073** for the entire period and **3.707** vs. **2.706** at the fifth semester, respectively).

On both community-based and ego network components, all previous investigations accord with the conclusions. The ego-network density test findings also indicate that successful students and their peers have strong relationships. These findings in our investigation support **RQ-NA3** and **RQ-NA4**.

Fixpoint Features: Our data show that at-risk students had higher fix-point values in general. For the same reason, at-risk students engage in numerous short-term relationships with others. In the early phases, the pattern is distinct (supporting research question **RQ-NA3**). For example, in the early stage of degree centrality, successful students' degree values are more significant than at-risk students' degree values. On the other hand, at-risk students already have higher fix-point scores. Therefore, students who have been exposed to too many early connections can be identified. The $p-value$ in each case is less than 0.01.

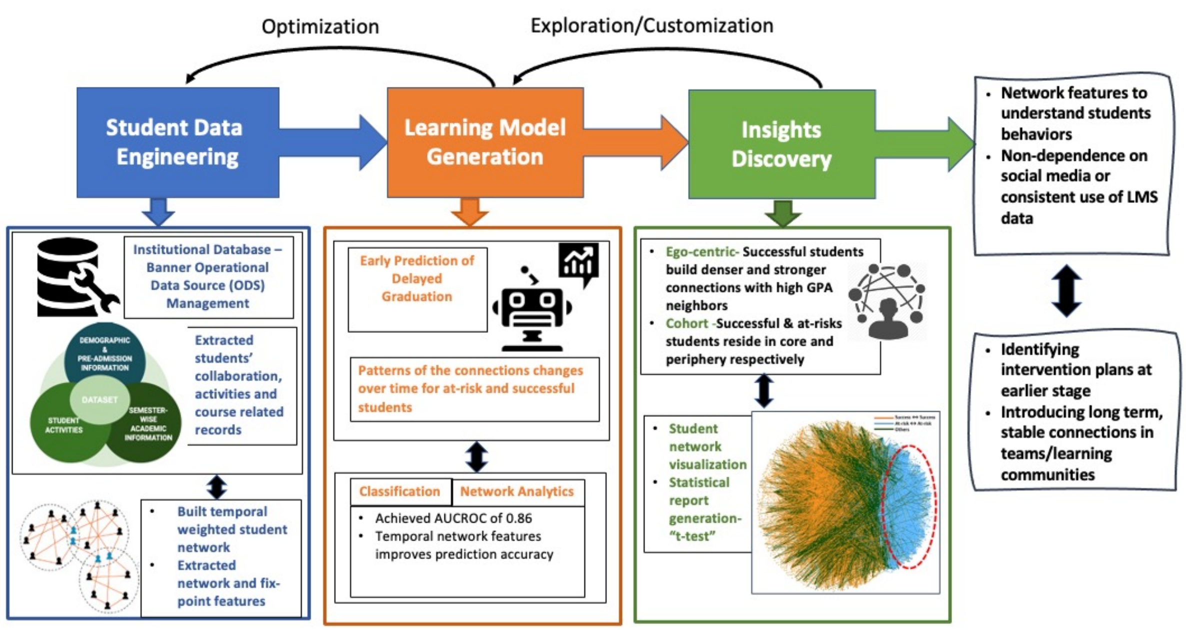


Figure 5.2: FIND analysis on student network model [1]

5.1.5 Insights Discovery

Figure 5.2 demonstrates the three main building blocks of FIND analysis on the student network model. In this section, we summarize the results of our network analysis model. We start by presenting some high-level findings and then delivering the insights learned from this model.

High-level Findings of Network Analysis:

1. The top half of at-risk students have higher centrality values than most successful students, showing that this group of at-risk students have access to more interactions.

2. The bottom half of low-performing students have a smaller value of degree centrality than the successful students, showing that many at-risk students do not connect with their peers as strongly as successful students.

3. The average degree centrality of successful students in the fifth semester is higher than the average for the entire period, implying that:

   (a) During early academic times, successful students can quickly steady their associations with other students; and

   (b) In later academic years, successful individuals do not develop many new relationships.

4. Successful students have larger mean community feature values than the at-risk students, indicating that they have a more significant impact on their communities.

5. In comparison to successful students, at-risk students had higher average values for betweenness centrality. It means that they mainly participate in forming bridges and live in the periphery of communities rather than core members.

6. Students at risk build connections with many groups and, as a result, end up on the margins of the community, making them more likely than successful students to be members of significant communities.

7. The Ego Networks of successful students are more intact than those of at-risk students, implying that successful students and their peers have deep bonds.

The following are conclusions drawn from network analysis:

1. Students at risk form haphazard connections, but successful students maintain and deepen existing ones from the beginning until their graduation.

2. Successful students are connected with neighbors who have a higher average GPA, and their bonds are stronger as a result.

3. The Ego-networks of successful students maintains a steady density over time.

4. Some at-risk students continue to form new connections even after graduation, and their ego networks do not become dense or complete.

5. At-risk students are on the outskirts of the communities, while successful students reside in the central area of the communities.

5.1.6 Lessons Learned and Limitations

Finally, this study concludes that incorporating students' network and fix-point features in the analysis makes it simpler to find successful intervention spots for at-risk students at an early stage. Consequently, successful and at-risk students exhibit different behaviors over time. If the policymakers or faculty advisers can alter or intervene in the way students connect, they may make a difference in graduating them on time. The significant challenges and limitations faced while developing the network analytical model are as follows:

1. Limitations of the Student Network model:

(a) Extracting and defining the rules for determining the shared neighbors to build the student network,

(b) Prioritizing and defining weights to each rule for determining the network is as hard as extracting and clarifying rules. Since we define a student in many ways with varying features, finding out the critical rules to determine a cohort or a student individually is harder.

(c) Dealing with the sparsity of the data due to inconsistent usage of LMS or different nature of instruction and logging student and course information

2. Administrative and Strategic Limitations:

(a) Dealing with the changes in curriculum over the 13 years of data, such as changes in course names and design: This study requires a vast volume of data from a more extended period. Since the department rearranges and changes the curriculum within specific periods, it isn't easy to aggregate over the course information.

(b) Unavailability of some critical student demographic information.

(c) Gender, race, ethnicity, generational social class, information on financial aid availability, student body demographics, institution's geographic location, and students' socioeconomic position are all inadequate or missing. It is one of the criteria that go into determining how long a student takes to graduate.

5.2 Machine Learning Techniques for Performance (CGPA) Prediction

In the following two machine learning models, we will explore successful and at-risk students' cohorts using a threshold of cumulative grade point average(CGPA) of 2.00. Section 5.2.1 demonstrates an unsupervised machine learning approach with K-means clustering to determine segregated-temporal patterns among successful and at-risk students. A supervised machine learning model with Long Short Term Memory

Recurrent Neural Network (LSTM RNN) is implemented to predict CGPAs ahead of time (See section 5.2.2). Chapter 6 demonstrates some eXplainable AI (XAI) techniques and tools to understand the training and prediction results since RNN is a black-box model and hard to interpret.

5.2.1 Unsupervised K-Means Clustering

This section explains how to use unsupervised machine learning to evaluate student temporal data models and locate clusters of students with similar success and risk patterns [5]. For constructing the feature vectors as input to the K-means clustering, the engineered features (see section 4.2.2) and semester-level nodes are only applied.

5.2.1.1 Problem Formulation

The clustering was applied only to the students who have at least eight semesters of information. In this analysis, students with a CGPA greater than or equal to 2.00 are considered successful, while others are at-risk. Several engineered features are generated, and the clustering algorithm is applied to them. However, the algorithm cannot find distinctive and valuable patterns in all of them. Seven significant engineered features are selected (See section 4.2.2.2) that capture the student data's temporal pattern and have a high correlation with the success criterion (e.g., CGPA set to threshold 2.00). This analysis utilized K-means clustering [126], an unsupervised learning approach. The dataset is strongly skewed toward successful class. Unless the model overfits this type of imbalanced data, supervised learning does not provide high accuracy. As a result, an unsupervised clustering method is utilized to identify patterns of student behavior.

The data for each aspect in the semester-by-sequence model serves as the foundation for analyzing students' progress in a major. This dissertation offers a method for mapping each engineered complicated feature in the sequence model to more straightforward signatures (re-representation) and extracting metadata from the signatures for unsupervised cluster analysis. A 2D feature vector(see Figure 4.5) is created for each student's enrolled semesters at the beginning of the analysis. Then, for all students who have been enrolled for at least eight semesters, K-means clustering is applied [126] for each of the seven selected engineered features. However, to acquire the best clustering results with the k-means clustering algorithm, it is necessary to know the optimal number of clusters to initialize the cluster centers and clustering algorithm.

The K-means clustering method is an iterative procedure that attempts to split a dataset into K separate non-overlapping subgroups (clusters). Each of the data points belongs to one single cluster. It aims to make intra-cluster data points as similar as possible while keeping clusters distinct (far) from each other (See Figure 5.3).

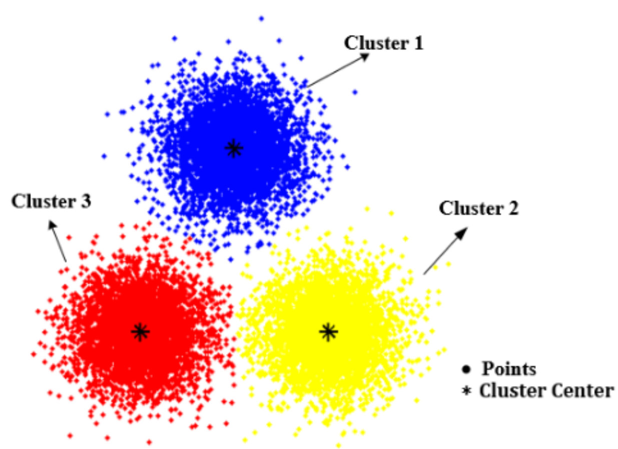


Figure 5.3: Concept diagram of k-means clustering [6]

The elbow method is a heuristic for determining the number of optimal clusters in a data set in cluster analysis. The technique entails plotting the explained variation as a function of cluster number and selecting the curve's elbow as the number of clusters to use (See Figure 5.4). The K-means algorithm is initialized with that number of clusters. As a result, the Elbow technique [127] estimates the appropriate number of clusters by clustering inertia before executing the k-means clustering. Each cluster is examined to see any discernible pattern for the groups of students (i.e., those classified as successful or at-risk) based on the success criterion.

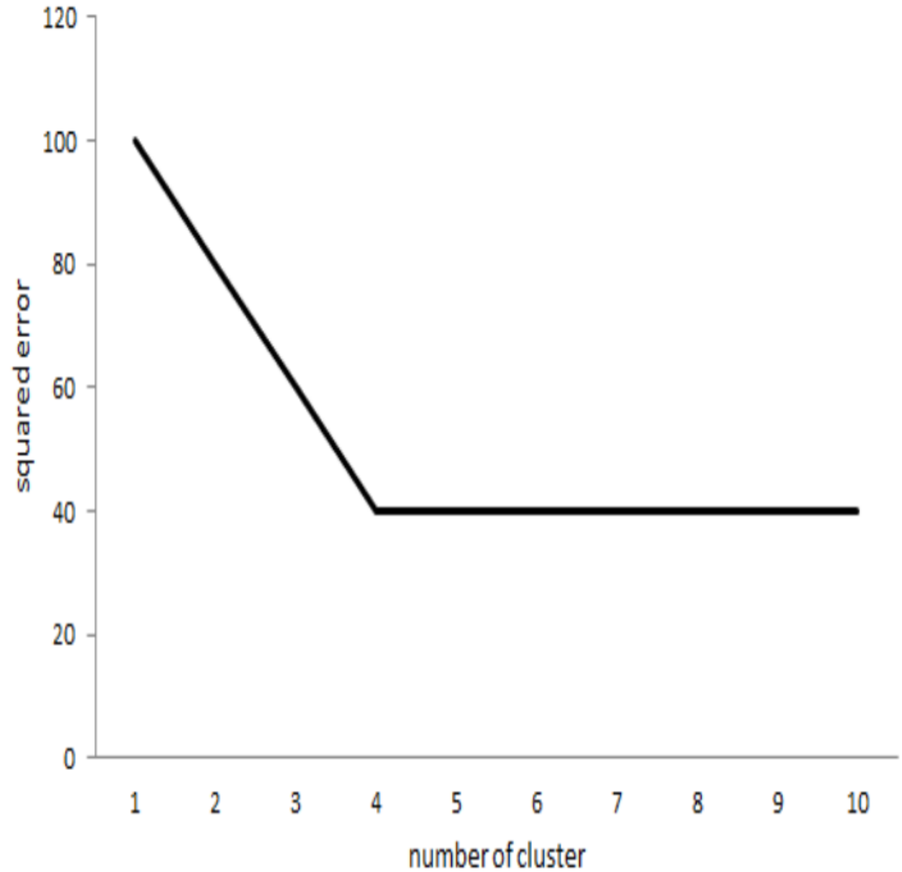


Figure 5.4: Elbow Point [7]

EAGER Learning Analytics tool utilizes this unsupervised K-means clustering for user study purposes (See Appendix A.2). In the user study, participants such as the academic and faculty advisers gave feedback on the tool and the analytics. There was a plan to include the students they are advising at the time of the user studies. However, the analysis consists of only the students who have at least eight semesters of data. Many students are enrolled for less than eight semesters. A similarity metric called DTW is extensively used in speech and word recognition [128] to solve this issue. Dynamic Time Warping calculates the similarity or distance of two arrays or time series of various lengths (See figure 5.5). It calculates the pointwise distance for two unequal time series vectors.

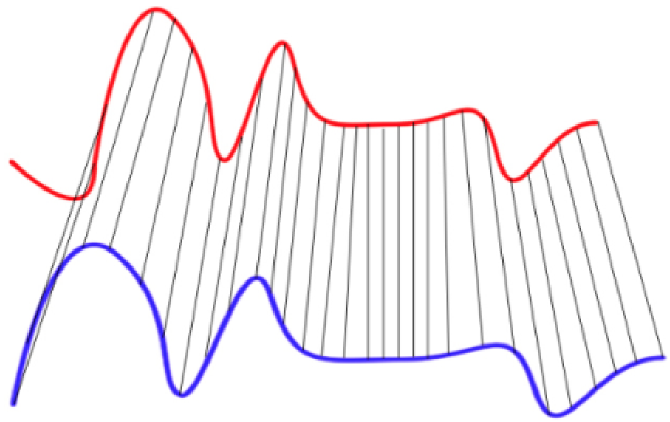


Figure 5.5: Dynamic Time Warping Matching [8]

Suppose for a particular engineered feature there are three optimal clusters extracted by the K-Means clustering. In that case, the similarity of a student having less than eight semesters is calculated with each cluster iteratively. The smaller the distance of the feature vector of a student to a specific cluster, the higher the chance that the student belongs to that cluster. Then whichever returns the lower distance, the student is more similar to that cluster.

5.2.1.2 Basic Steps

In summary, the basic steps for the unsupervised cluster analysis are:

- Step 01: Forming a 2D feature vector for each of the features wherein the feature vector
  - Each column represents a student.
  - Each row represents the value of the feature in a semester.
- Step 02: Using k-means clustering on the engineered features
  - Applying the Elbow method for determining the optimal number of initial clusters for K-means clustering

- Step 03: Considering seven significant engineered-temporal features which present meaningful temporal and behavioral patterns for successful and at-risk students.

- Step 04: Analyzing the behavior of each cluster for each of the engineered temporal features

  - Finding the students in the optimal clusters who are enrolled in less than eight semesters with a similarity measure (Dynamic Time Warping - DTW)

  - Insight discovery for planning proper intervention

5.2.1.3 Example Cluster Analysis on Engineered Features

Extracting metadata from each cluster of students demonstrates the analytic process, allowing the researchers to identify the groups of similar students and outliers. The cluster analysis results are discussed on two examples of engineered features among the seven: 1) Course Progression Through Semesters and 2) Progression of Good Academic Standing through Semesters.

For the feature "Course Progression Through Semesters," the K-means clustering technique identifies three suitable clusters in Figure 5.6. If the students are performing well, the value of this characteristic should be steadily growing.

Following are the statistics of the clusters generated by the K-means clustering algorithm -

- the blue cluster has a total of 483 students where 90.27% of the students are successful

- the green cluster has a total of 616 students where 84.58% are successful students, and

- the orange cluster represents a higher number of at-risk students, where 72.21% of the total 367 students are successful.

The successful group is the most dominating cohort in the blue cluster, with 483 students in total. 90.27% of the group's members are successful. Each semester, this cluster has a higher course level value than the other two clusters, and the feature value continuously increases until the eighth semester. Thus, this set of students appears to have registered in courses with a higher number of course levels. A small percentage of at-risk students belong to this high-achieving group (only 9.73%). The green-colored cluster has a higher rate of course level progression than the orange-colored cluster. The orange cluster has a lower average value of course levels that do not cross 2000 level courses, even until the eighth semester. Students in the orange and green clusters follow a similar pattern until the second semester. Students at risk are higher in number in the orange cluster than the other two clusters. The course evolution through semesters exhibits distinct patterns of student groupings based on our success criterion.

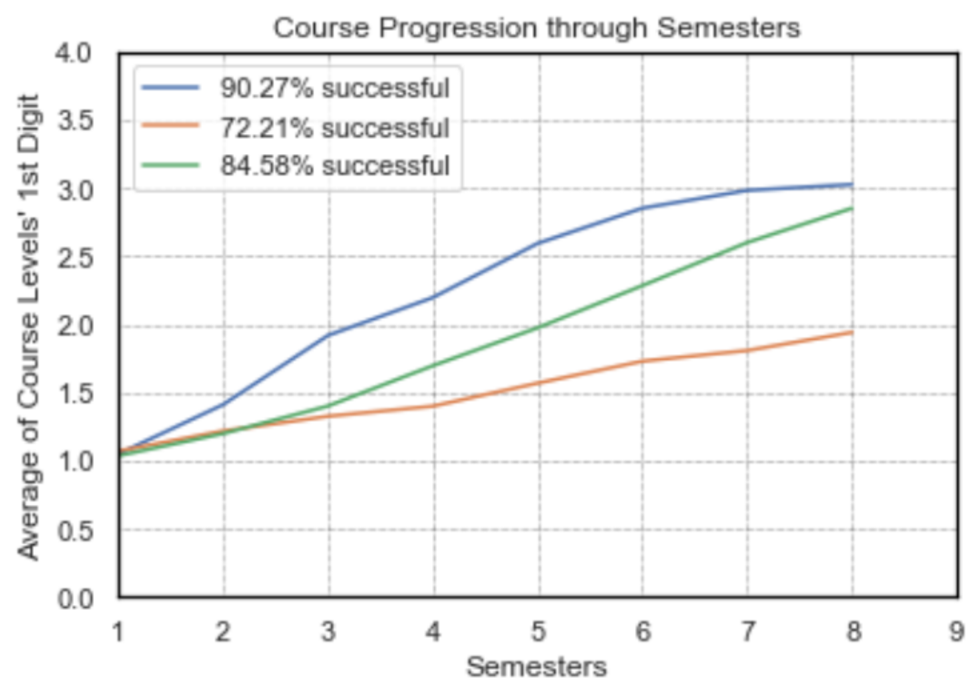


Figure 5.6: K-means Clustering Analysis on Course Progression over first Eight Semesters [5]

A second engineered feature is the mean of good academic standings over the first eight semesters, which depicts an identifiable correlation of groups of students with the target criterion (See Figure 5.7). The statistics of the clusters as follow:

- the blue contains 1262 students in total, where 14.18
- the green cluster has a higher number of at-risk students where they are 33.10% of the total 145 students, and
- the orange cluster has 28.81% at-risk students of a total of 59 students

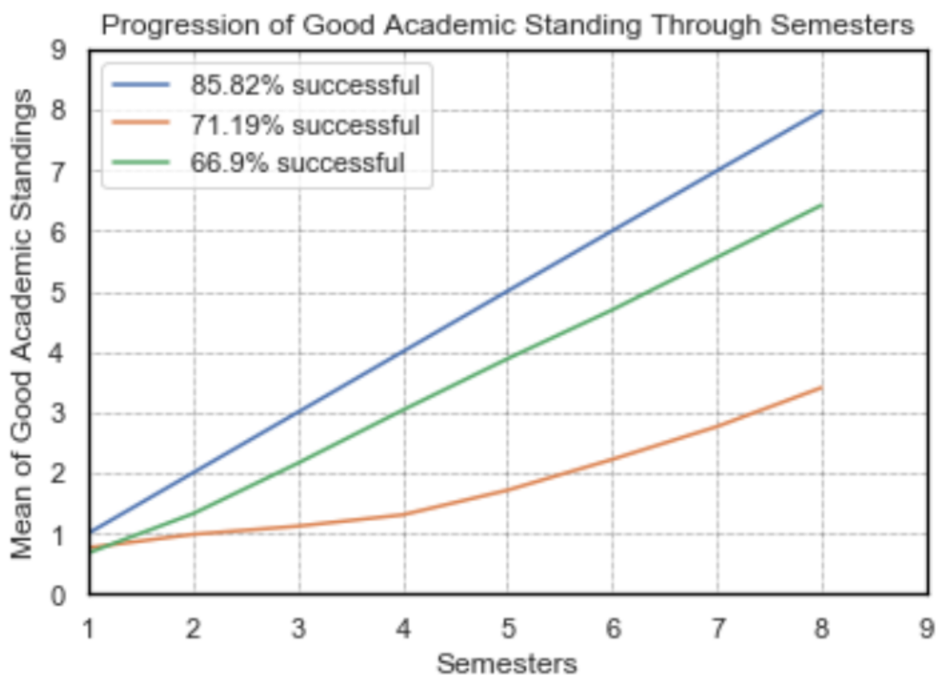


Figure 5.7: K-means Clustering Analysis on Progression of Good Academic Standing over First Eight Semesters [5]

The blue cluster, which successful students dominate, has a value that rises steadily until the eighth semester. As a result, a successful student is nearly sure to be in good academic standing each semester. However, where 28.81% are at risk, the orange cluster is in good academic standing only once until their fourth semester. In this cluster, the students are in good standing for a maximum of three times, even until the eighth semester.

Figures 5.6 and 5.7 show how the K-means clustering method produces a new representation of the temporal sequences that distinguish successful students from at-risk students who do not have a current CGPA higher than 2.00. This new form is used as the foundation for extracting metadata (features) that provide insight into a student's achievement and progress throughout the first eight semesters.

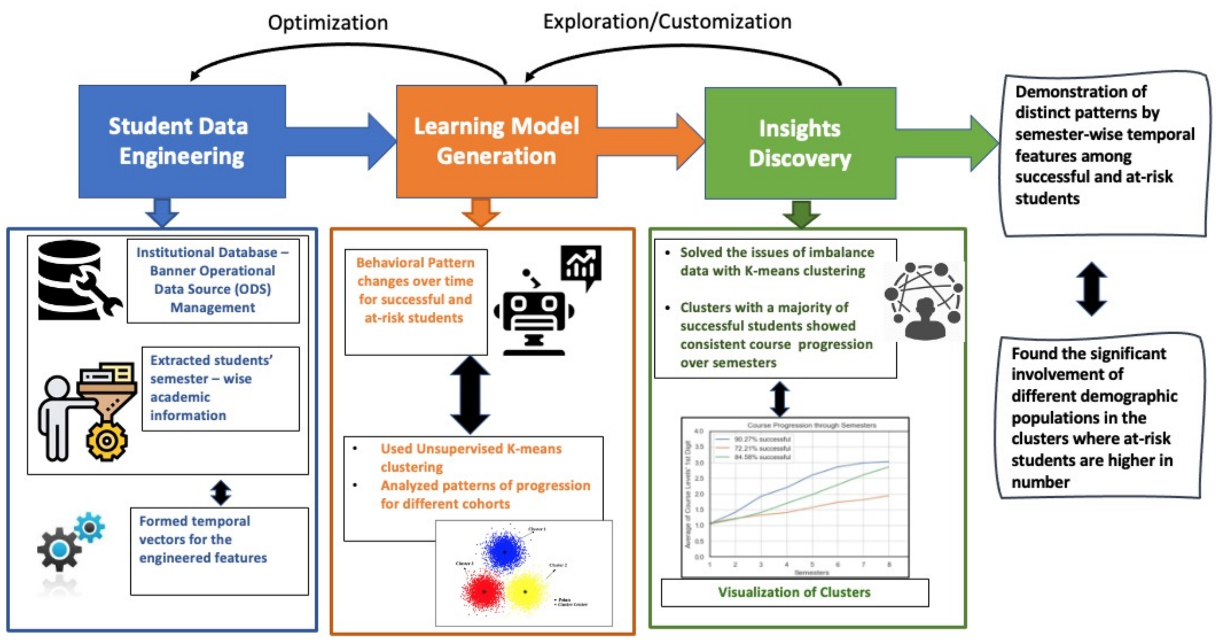


Figure 5.8: FIND on Unsupervised K-means Clustering on Temporal Data [5]

5.2.1.4 Insights Discovery

Figure 5.8 shows the three main building blocks of the unsupervised K-means clustering on the students' temporal features. The findings from the unsupervised learning model led to some insights that can be leveraged to plan intervention.

The main takeaways drawn from this unsupervised learning model are two-folds:

- Throughout their enrollment, the models can take into account the students' timewise interdependence. Moreover, it allows for greater freedom when determining the duration of a temporal node, contextualizing information within a node, and understanding node sequences through time, and

- The complex and heterogeneous student data can be mapped into a set of disjoint clusters, making it easier to analyze while also separating successful and at-risk students. As a result, it is easier to predict students at risk and plan proper intervention tips for them early.

The findings of the unsupervised model derive the following insights from the student data:

- The engineered features in the clusters show a proportional association between students' overall performance as successful or at-risk cohorts. With an increasing number of at-risk students in particular clusters, performance suffers (e.g., the orange clusters in Figure 5.6).

- In this case, students' academic success is projected using normalized and temporal course-level progression vectors. The k-means clustering analysis results reveal three distinct performance patterns (See figure 5.6): 1. **Consistently and rapidly** increasing course level progression with almost 100% of successful students in the blue cluster, 2.**Consistently and linearly** increasing course levels with most successful students in the green cluster, 3. **Flattened and inconsistently** increasing course levels with most of the at-risk students of the dataset present in the orange cluster. Tracking such characteristics from the start of the enrollment period and mentoring students can help students perform better from the beginning.

5.2.1.5 Lessons Learned and Limitations

The limitations from this unsupervised learning model are three-folds. First, identifying the engineered features that demonstrate a fine line between the characteristics of successful or at-risk students is challenging. The model has been examined with a set of seven features presented in section 3.2.2. However, only a few features (shown in Figures 5.6 and 5.7) were able to demonstrate that fine line between the characteristics of successful or at-risk students. Second, there is still a problem with the imbalanced data since all the clusters have the most successful students. There is no significant involvement of different demographic populations in the clusters where at-risk students are higher in number and co-reside with many successful students. Finally, finding the most likely clusters for students with a small number of semesters (less than eight semesters) is challenging. Although DTW is handling the issue, it might not be the most efficient method since this is the only way the similarity is being extracted. One similar cluster has been selected with this method, where the distance is the smallest among all other clusters even though the smallest distance might be an enormous value. It is necessary to navigate other similarity measures for this analysis.

5.2.2 Long Short Term Memory Recurrent Neural Network for Supervised Learning

This section will discuss a deep neural network for time series analysis named Long Short Term Memory (LSTM) Recurrent Neural Network (RNN). The analysis was formulated with both classification and regression analysis based on CGPA.

5.2.2.1 Application of a Vanilla LSTM RNN for Classification

A recurrent neural network model was developed for the early prediction of at-risk students based on CGPA. The target variable for the supervised classification model was to predict students with a CGPA of more than 2.00 and vice versa. This section starts with the definitions and working mechanism of fundamental recurrent neural networks and LSTMs.

In an RNN, the training process involves a loop where information cycles. RNN adds the nearest past to the present input. It considers the current input and what it has learned from the inputs it received earlier. Therefore, a Recurrent Neural Network has two inputs, the present, and the recent past. The data sequence contains crucial information about what is coming next, so an RNN can do things other algorithms can not. For a general feed-forward network, outputs are independent of each other. The output at the time t is independent of the output at time $t-1$.

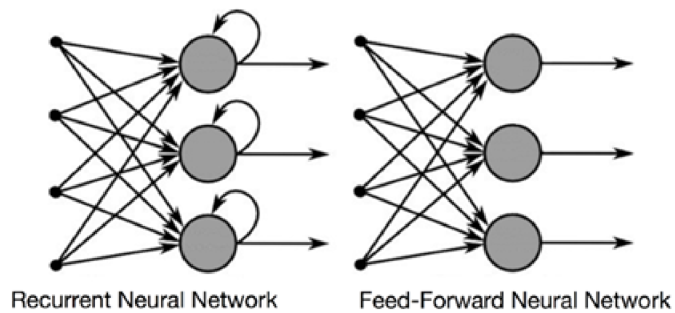


Figure 5.9: Recurrent Vs Feedforward Neural Network [9]

A primary RNN input is a 3D NumPy array, where it can be represented as such [samples, time steps, features]. The 3D NumPy array consists of several 2D NumPy arrays, where the number of 2D arrays is the number of instances, meaning in our case, it will be the number of students. Therefore, the input array to RNN contains a 3D array which consists of N number of 2D arrays, where N= total number of students (samples). Furthermore, each 2D array, i.e. each student's data, consists of rows and columns, rows represent semesters (which is ordered by beginning to current/last semester if graduated), and columns represent features (See Figure 5.10).

A recurrent neural network cell contains feedforward neural networks because the data includes time steps for each sample (in our case, each student). The RNN has four main elements: a sequential input, a sequential output, multiple timesteps, and multiple hidden layers. The hidden states in the hidden layers are updated after each prediction, implying that the prediction depends not only on the input values but also on the input sequence context and earlier hidden state values. After one single pass in the feedforward step, the model calculates the forward and backpropagation error. "Cross-entropy" is used as the cost function to calculate the error and update the neurons' weights accordingly.

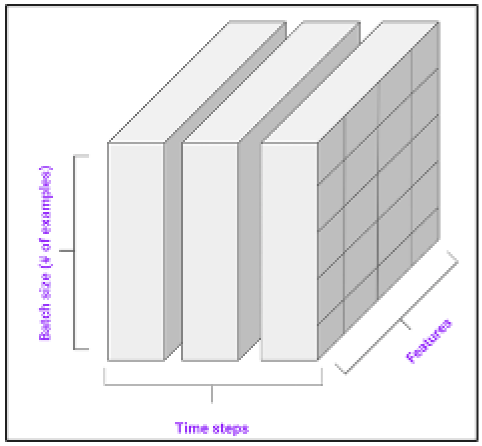


Figure 5.10: Input Data Representation => [samples,time-steps,features] [10]

In a simple RNN, the final output nodes of the network are less sensitive to the input at time $t(semester) = 1$. The vanishing gradient problem causes this issue which makes the gradient-based learning (GBL) process difficult. In GBLs, in each iteration of training, each of the neural network's weights receives an update proportional to the partial derivative of the error function for the current weight. During the weight update process, in some cases, the gradient may become vanishingly small. Therefore, it prevents the weights from changing their values. This scenario may ultimately stop the neural network from further training and learning. The fading of the color throughout the layers in figure 5.11 presents the fading effect of the gradients. Long Short Term Memory (LSTM), a special kind of RNN, solves this issue. LSTM is considered to avoid the problem of vanishing gradients in RNN. Theoretically, the information in RNN is supposed to follow for arbitrary large sequences, but this does not hold up in practice. The information starts to vanish due to a lower value of gradient.

LSTM networks utilize smarter memory blocks for preserving recent sequences. Each of these memory blocks contains three types of gates that manage the state and

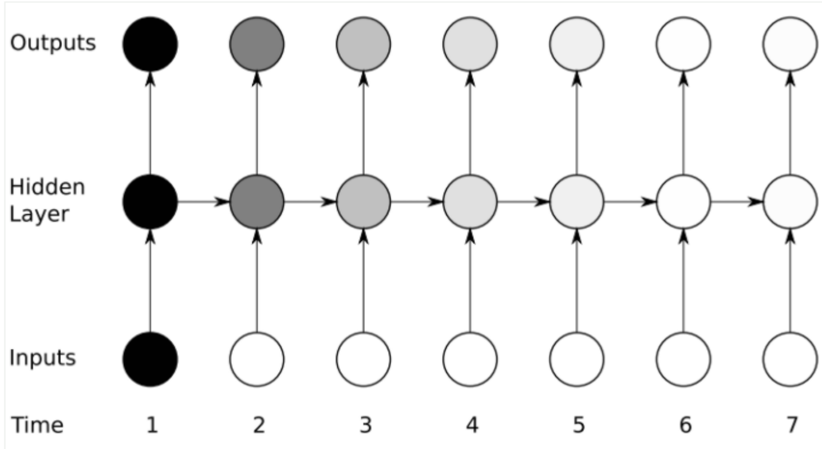


Figure 5.11: The vanishing gradient problem for RNNs [11]

output of the sequence - forget, input, and output gates. Each gate inside a block uses the sigmoid activation units to control whether or not they are triggered, making the change of state and addition of information flowing through the block conditional. Each unit is similar to a mini-state machine, with weights learned during the training phase on the gates. It's easy to understand how a layer of LSTMs could be used to achieve complex learning and memory, and it's not difficult to conceive how higher-order abstractions could be built with numerous such layers.

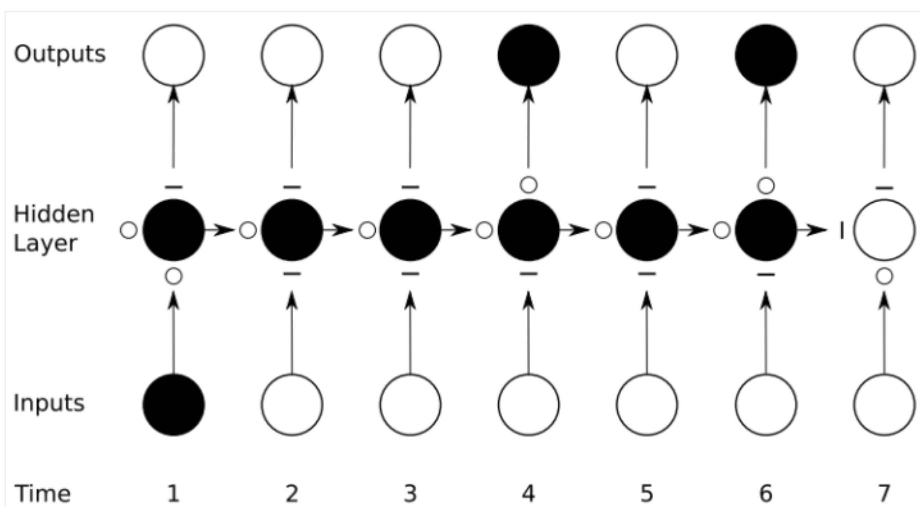


Figure 5.12: Preservation of gradient information by LSTM [11]

As the first version, a vanilla Encoder-Decoder LSTM RNN consisting of two recurrent neural networks (RNN) is implemented that act as encoders and decoder pairs (See Figure 5.13). The encoder maps a variable-length source sequence to a fixed-length vector, and the decoder maps the vector representation back to a variable-length target sequence. The input format to the LSTM model is $[samples, time - steps, features] -> [None, None, features]$. The student data have varying numbers of semesters. A "None" dimension in a shape tuple means that the network will accept inputs of any number, which will resolve the problem of fitting the data with a varying number of semesters. The last layer predicts the values of CGPA with a softmax activation function, and then it calculates if it is below or above the CGPA threshold (CGPA 2.00).

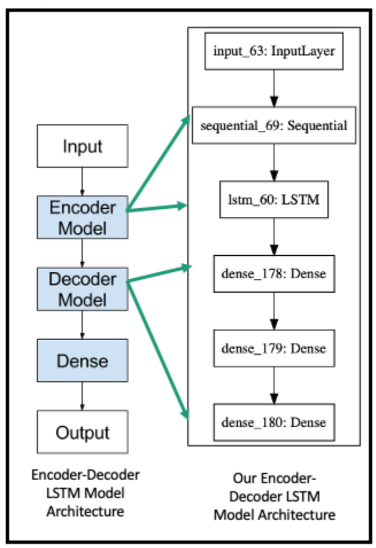


Figure 5.13: Vanilla Encoder-Decoder LSTM Network

5.2.2.1.1 Data Preprocessing and Hyper Parameter Tuning

For this analysis, we use the same dataset with 6220 undergraduate students as the unsupervised clustering analysis. Here is a list of data pre-processing steps that are followed:

1. Pre-processing of the data, which includes

(a) Selecting salient features: Multiple salient features are chosen based on multiple rounds of discussions with the domain experts and prior work on this project [3].

(b) Removing sparse features: Sparse and inconsistently recorded features are removed. If the feature information is available in less than 50% of records, they are considered sparse features and removed from the analytics.

(c) Adding engineered temporal features: The analytics only include engineered features from the course information. Later, in the analysis, it is found that these features are directly correlated to CGPA calculation. Therefore, the appendix does not include these features since, ultimately, they are discarded from the final analysis. Such features are -

i. Total number of F

ii. Total number of D

iii. Total number of C

iv. Total number of B

v. Total number of A

vi. Total number of course withdrawals

2. Scaling the features using "Min-Max" scaling. In this technique, values are shifted and re-scaled so that they end up ranging between 0 and 1.

3. Applying one-hot encoding to the categorical features

Batch-wise Model Building: The data have students enrolled for varying numbers of semesters. A recurrent neural network can handle various semesters. One single model is trained with different batch sizes and different timesteps in the data without the need to frame the missing timesteps with "zero paddings." In RNN, the batch size for training means how much data the model will train for each epoch. For example, if there are 1000 rows of data and the epoch is 100 and batch size is 10, then there will be $1000/10 = 100$ batches for every single epoch, each with a length of 10 to train the model. The RNN model is initialized with NONE-type timesteps. So it will take data with any number of timesteps. The process loops through one to the highest number of semesters the data has, and in each iteration, the model is trained with that particular set of data. For example, when the loop counter is 1, the model is trained with all the students' data in only one semester. This data contains students with a maximum of 24 semesters. Hence, the model is trained for one semester of data to a maximum of 24 semesters.

5.2.2.1.2 Results and Discussion with Vanilla LSTM RNN

The model is trained on both ongoing and graduated students with a validation split of 70:30. The following two tables present the accuracies, false positives, and negatives for the first ten batches of data with and without the categorical features for CGPA threshold 2.00. The model achieves a 95.86% accuracy with both the categorical and numeric data and a 98.98% accuracy with only using numeric data. In Table 5.3, the number of false positives and false negatives are relatively high. The data is highly imbalanced, and most of the batches (#semesters) have a percentage of successful students around 70% to 85%, which is nearly the accuracy of the model for that particular batch. Therefore, the model is only learning the pattern for the majority class and failing to predict the minority class, which is the students at risk. As for a deep learning approach, the model is using a very minimal number of samples. Therefore, more data will improve the model's accuracy. It would be easier to observe the learning (fitting) or unlearning (overfitting or underfitting) process of the model with more data.

Figure 5.14 demonstrates that the training loss is lower than the validation loss, and the validation loss fluctuates. This observation suggests the model is overfitting.

Table 5.3: Performance records with both categorical and numeric data

# semesters	Accuracy (%)	F1-score	False +ve	False -ve	Successful: At-risks in training dataset	Successful: At-risks in test dataset	Total # students (Successful: At-risks)
1	87.31	1	17	20	438:145 (75:25)	173:78 (68:32)	834 (76:24)
2	90.91	1	10	5	201:118 (63:37)	80:58 (57:43)	458 (62:38)
3	92.13	0.959	6	14	339:55 (86:14)	150:19 (88:12)	564 (87:13)
4	93.98	1	8	10	230:36 (86:14)	99:16 (86:14)	381 (86:14)
5	93.70	0.987	3	10	308:41 (88:12)	136:14 (90:10)	499 (89:11)
6	90.66	1	4	12	238:19 (92:8)	100:11 (90:10)	366 (93:7)
7	94.90	1	4	8	328:25 (93:7)	143:9 (94:6)	505 (94:6)
8	95.86	0.995	8	13	437:22 (95:5)	182:15 (92:8)	656 (95:5)
9	89.40	0.975	5	14	333:35 (90:10)	145:14 (91:9)	529 (91:9)
10	85.93	0.993	2	18	307:27 (92:8)	132:12 (92:8)	478 (92:8)

Even though the training loss is lower than the validation loss, the training loss is still too high, and it is not improving over the epochs.

5.2.2.2 Major Takeaways from the Vanilla LSTM RNN and Change of Direction

The primary usefulness of this model is this model can be used in real-time production with a selected number of features. This model can deal with students' data with varying lengths of semesters. Some limitations of the model are listed below-

- By the predictive results and interpretation of the training and predictive results analysis (See section 6.2.1), it is hard to conclude why the model reports a higher number of false positives and false negatives or if the model is overfitting.

Table 5.4: Performance summary with only numeric data

# semesters	Accuracy	F1-score	False +ve	False -ve	Successful: At-risks in training dataset	Successful: At-risks in test dataset	Total #students (Successful: At-risks)
1	99.60%	0.996	0	1	438:145 (75:25)	173:78 (68:32)	834 (76:24)
2	94.93%	0.9493	3	4	201:118 (63:37)	80:58 (57:43)	458 (62:38)
3	94.71%	0.9471	5	4	339:55 (86:14)	150:19 (88:12)	564 (87:13)
4	98.26%	0.9826	0	2	230:36 (86:14)	99:16 (86:14)	381 (86:14)
5	96.67	0.9667	3	2	308:41 (88:12)	136:14 (90:10)	499 (89:11)
6	98.18%	0.9818	1	1	238:19 (92:8)	100:11 (90:10)	366 (93:7)
7	98.68%	0.9868	2	0	328:25 (93:7)	143:9 (94:6)	505 (94:6)
8	98.98%	0.9898	2	0	437:22 (95:5)	182:15 (92:8)	656 (95:5)
9	97.48%	0.9748	3	1	333:35 (90:10)	145:14 (91:9)	529 (91:9)
10	97.22%	0.9722	0	4	307:27 (92:8)	132:12 (92:8)	478 (92:8)

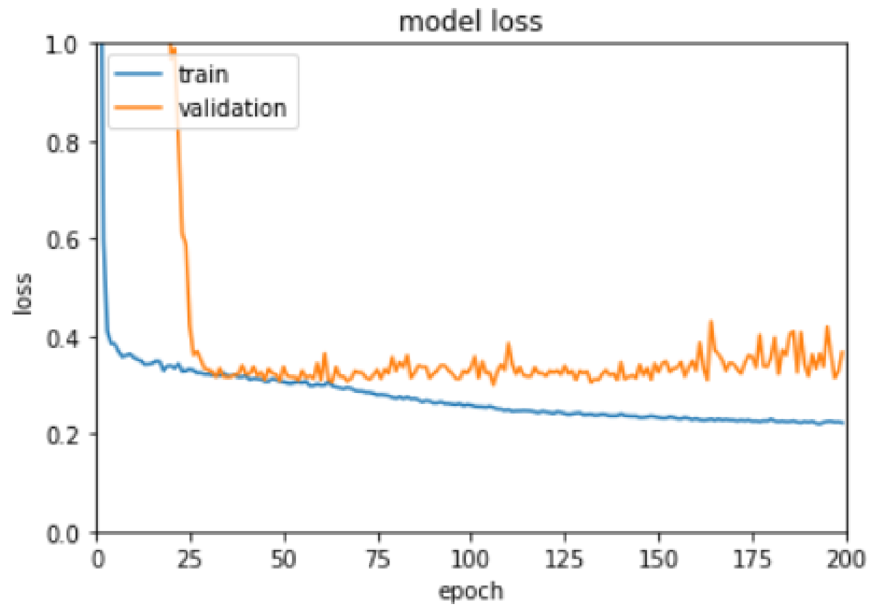


Figure 5.14: Diagnostic plot of Vanilla LSTM RNN showing an overfitting model

- Moreover, a small number of samples is used, which is inadequate to train a deep neural network. Therefore, it may be possible to achieve higher accuracy, lower false confidence, and better interpret the model and the results by including more data.

- The model also lacks different analytical insights that involve learning analytics research. For example, it is necessary to incorporate courses, major CS courses to observe how the analytics work for different cohorts (Male vs. Female, freshman vs. transfer, etc.).

- Moreover, the domain experts in our research group suspect that the model still manages to achieve higher accuracy since some of the features are directly correlated to the calculation of the CGPA. Examples of such features are academic standing, quality points, the total number of A, B, etc.

- All the researchers involved in this learning analytics group agreed that the batch calculation might not be proper. In this analysis, the first batch of students is the students who are only enrolled for one semester. But everyone in the dataset is at least enrolled for one semester. The analysis is missing those data for training the model.

The analysis will be improved based on these observations on the limitation-

1. A cumulative approach was incorporated to increase the number of samples or data and proper batch calculation. For example, all the students enrolled for at least one semester are considered for the first batch of the semester.

2. The features which are directly correlated with the CGPA calculation were excluded. Alternatively, engineered, temporal and normalized features such as "percentage of A Credits," "percentage of B credits," "course level progression" are reutilized from the unsupervised cluster analysis.

3. Only the model with both categorical and numeric data is considered.

4. Heavy lifting of hyperparameter tuning was done with different types and values of regularization. The best model was chosen by grid search. Here are some details on the regularization. The elastic net with input weight regularization [129] gives the best performance. The results are reported in table **??**.

   (a) $L1$ and $L2$ regularization are used with various values -

      i. $L1L2(0.0, 0.0)$ [e.g. baseline]

      ii. $L1L2(0.01, 0.0)$ [e.g. $L1$]

      iii. $L1L2(0.0, 0.01)$ [e.g. $L2$]

      iv. $L1L2(0.01, 0.01)$ [e.g. $L1L2$ or Elastic Net]

   (b) Different types of regularization are explored -

      i. Recurrent weight regularization

      ii. Input weight regularization

iii. Bias weight regularization

5. The oversampling method SMOTE [130] was used since the data is highly imbalanced. It up-samples the minority class in the training dataset - so the ratio of successful and at-risk students in the training dataset is 50 : 50.

Table 5.5: Performance records with regularization and oversampling on cumulative data

# semesters	Accuracy	Precision, recall and F1-score	False +ve	False -ve	Successful: At-risks in training dataset	Successful: At-risks in test dataset	Total # students (Successful: At-risks)
1	82.52%	1	247	568	93:7	92:8	6220
2	82.60%	1	106	387	93:7	94:6	5386
3	82.34%	0.9588	83	231	93:7	94:6	4929
4	82.49%	1	71	200	93:7	94:6	4366
5	81.06%	0.9867	57	136	94:6	93:7	3985
6	81.15%	1	57	95	93:7	94:6	3486
7	80.48%	1	40	129	94:6	94:6	3118
8	82.38%	0.9949	44	89	94:6	92:8	2613
9	81.85%	0.9748	30	119	93:7	93:7	1957
10	79.43%	0.9931	27	121	94:6	93:7	1430

With the oversampling, regularization, and cumulative approach, the batch sizes are much larger this time (See Table 5.5. However, the data imbalance induced is much higher than before. Moreover, there is a higher number of false positives and false negatives. By using the oversampling method, the process is upsampling the minority class. In this case, those are the students at risk, which induces higher false negatives in the data. Since the oversampling introduces replicated at-risk students, it is discarded from the next iteration. The next version is a bi-directional LSTM RNN for a regression model discussed in section 5.2.2.3.

5.2.2.3 Application of a Bi-directional LSTM RNN for Regression Analysis

The information stored by the unidirectional LSTM is limited to the past. They only loop through the past from the future. Using bidirectional will run the inputs in two directions, one from past to future and the other from future to past. What distinguishes this approach from unidirectional is that information from the future is preserved in the LSTM that runs backward. The information flows from both past and future by using the two combined hidden states at any point in time. Therefore, BiLSTMs may produce excellent results since they are better at understanding the context from both sides.

5.2.2.3.1 Data Preprocessing and Heavy Hyper Parameter Tuning

In addition to the hyperparameter tuning discussed in the previous section, different types of LSTM were explored for the regression model with a "relu" activation function in the output layer, such as-

- Vanilla Stacked LSTM (encode-decoder)
- CNN LSTM with different types of pooling, filtering
- Bi-directional LSTM

By the grid-search with thousands of hyperparameter tuning, bidirectional LSTM produces the best results. Instead of the stacked decoder-encoder LSTM (See figure 5.11), bidirectional LSTM cells and the output layer with a "relu" activation function are used since this is regression analysis. Additionally, other hyperparameter tunings are variations of the batch size, different loss functions, optimizers, learning rates, drop-out layers, LSTM cells, early stopping, and different techniques to reduce validation and training loss.

Overfitting can occur when there are a lot of features. Apart from that, it will take longer to optimize hyperparameters and training methods in general. As a result, it is essential to initialize with the most relevant features. There are multiple techniques when it comes to feature selection. The "Univariate Feature Selection" is used [131]. Statistical tests (like χ^2) determine how strongly the output feature depends on each feature in the dataset. Python's sci-kit-learn library's $SelectKBest$ selects K best features using statistical tests (default is χ^2). A grid search on finding the best "K" for the feature selection is conducted. Please see the list of finalized features in appendix A.1.

This version of RNN tries to solve a regression problem. The model predicts CGPA and then classifies the output result to one of the following ranges of CGPA- $[2, 2.5, 3, 3.5]$. Therefore, there are five classes. Figure 5.15 shows a distribution of the bins of CGPAs. The x-axis presents the number of students in each category, and the y-axis represents the batches of the data, which is the number of semesters.

5.2.2.3.2 Results and Discussion with Vanilla LSTM RNN

LSTM RNN allows us to predict with multiple lookback windows. The analysis involves several lengths of lookback windows. Only the results with a lookback window of size one are reported here.

This time the analysis includes three more semesters with a total of 6466 students to increase the number of samples. From figure 5.15, it is clear that there is not that much class imbalance. However, the number of students having a CGPA of less than or equal to 2.00 is still lower than the other classes. For a standard classification or regression task, cross-validation is used to get a sense of the model's skill on new,

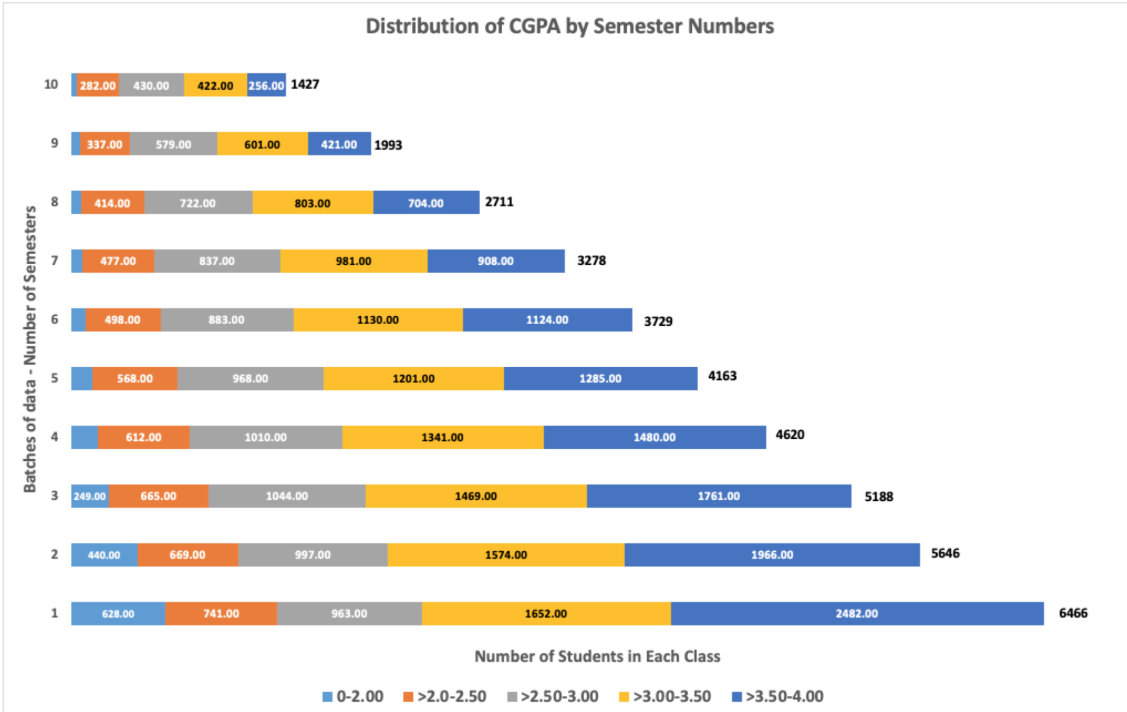


Figure 5.15: Distribution of the bins of CGPAs by number of semesters

unseen data. The order of the numbers is critical when working with time-series data. The ratio of train and test datasets is 70 : 30.

The LSTM models use the training history to diagnose the behavior of the model. Creating and examining these charts can help discover new setups to improve the model's performance. A model that performs well on the training dataset but poorly on the test dataset is **underfitting**. An **overfitting** model is one in which performance on the train set is good and improves over time, but performance on the validation set improves to a point before degrading. This can be seen in a plot where the training loss is lower than the validation loss, and the validation loss is trending upwards, indicating that more gains are feasible.

When the model's performance is good on both the train and validation sets, it is a good fit. To identify this, look at a graph that shows the train and validation loss dropping and stabilizing around the same position. Diagnostic plots are used to keep track of the validation and training loss. The *EarlyStopping* technique stops the model training if the validation loss starts to rise. The model was trained with 500 epochs for each of the batches. For most of them, the model does not halt due to the *EarlyStopping.* It is a good sign that the model is neither overfitting nor underfitting. Figure 5.16 shows the training and validation loss for the batch with all students until their fifth semester. The model starts to learn better and is doing well on both the training and validation dataset. The value of the validation loss falls to around 0.02. That is why the accuracy is relatively high reported in Table 5.6.

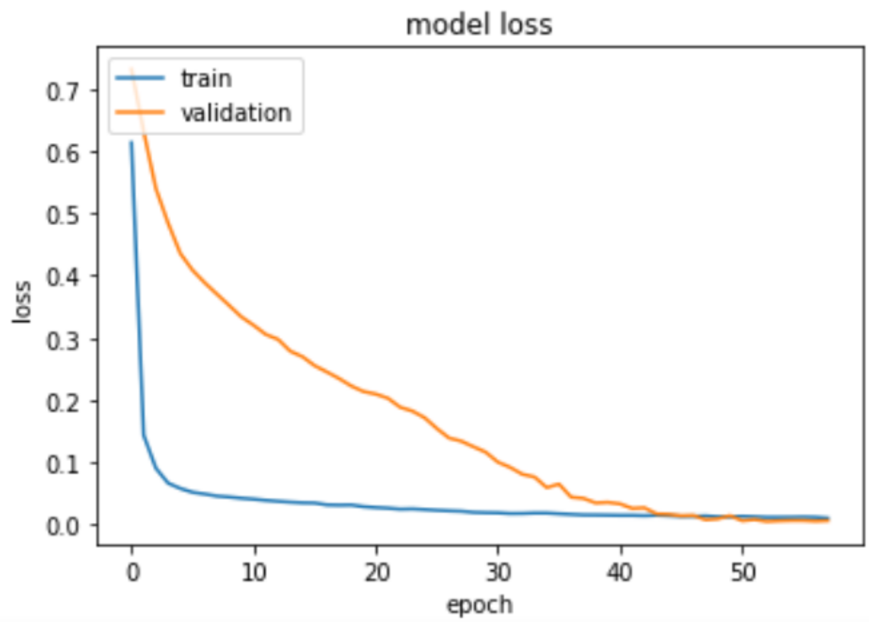


Figure 5.16: Diagnostic Line Plot Showing a Good Fit for a Model for number of semesters 5

The model achieved the highest accuracy of 92.16% for the number of semesters of four. The model also improved precision, recall, and F1-score other than the accuracy as the performance metric.

From the last versions, in this multiclass classification problem, the accuracy improved a lot. The class having a CGPA from 0 to 2.00 has many misclassifications for the first batch. It is 270 among 530 students. The number had decreased over batch sizes. Therefore, the model is not biased toward this imbalance. The proportion of these students is the same in all batches. For example, in the stacked bar chart in

Table 5.6: Prediction performance records with the Bi-directional LSTM

# semesters	Accuracy (%)	Precision	Recall	F1 score	# Misclassified	# missed in the class $0 - 2.00$	Total # students
1	91.76	0.8708	0.9176	0.8824	530	270	6466
2	91.80	0.8844	0.918	0.8863	460	190	5646
3	90.96	0.8613	0.9096	0.8772	518	60	5188
4	92.16	0.8493	0.92161	0.884	362	90	4620
5	91.82	0.8431	0.9182	0.879	340	170	4163
6	91.47	0.8367	0.9147	0.874	318	70	3729
7	89.96	0.8265	0.8996	0.8615	329	60	3278
8	90.58	0.8204	0.9058	0.861	255	30	2711
9	91.39	0.8352	0.914	0.8728	171	40	1993
10	90.98	0.8278	0.9098	0.8669	128	50	1427

figure 5.15, the "$0 - 2$" class is lower in numbers among all five classes. The misclassified instances also did not demonstrate any significant patterns or attributes to be concluded as the reason behind misclassification.

5.2.2.4 Findings and Observations

This dissertation involves multiple LSTM models with thousands of hyperparameter tuning. Findings from the observations -

- The volume of data samples plays a vital role in training a deep learning model. The first version of the RNN model demonstrates a low accuracy, high false positives, and false negatives. A diagnostic plot analysis also shows that this RNN model was overfitting.

- Moreover, the data imbalance problem does not get resolved if we do not have enough data to train. The model predicted the majority class, which means the model is overfitting on the majority class and is failing to predict the at-risk students.

- Model's performance is highly correlated to the features used in training. For example, the first version of RNN achieves almost 100% accuracy while using highly correlated features with CGPA calculation, biasing the model outputs. Removing such features increases the credibility of the model's accuracy. Secondly, LSTM cells are built for temporal analysis. Using non-temporal features with them may corrupt the structure of a time series model.

- Hyperparameter tuning with multiple options and grid search take an extended amount of time. But, in the end, finding the optimum model for a dataset pays off. The bidirectional LSTM achieves improved accuracy because of the meticulous hyperparameter optimization.

- Oversampling or undersampling may corrupt a model. Oversampling created a replica dataset, and the model overfits towards the replicated data. In the second version of RNN, oversampling causes higher false positives and false negatives for the student data (See Table 5.5).

- The same model can work better for a regression problem with tweaking in the activation function in the output layer. This is another implication of an imbalanced dataset.

Sections 6.2.1 and 6.2.2 explore both the classification and regression LSTM models with explainable AI tools. They focus on explaining the predictive results and training accuracy of the model in more detail.

CHAPTER 6: Interpretable Learning Analytics

This chapter first presents an interactive machine learning prototype that targets to help the domain experts to understand their students better in section 6.1. Then, a state-of-the-art explainable AI tool named SHAP is used to explain the classification model (discussed in section 5.2.2.1) built to predict student success and risk in section 6.2.1. SHAP [132] is an integrated visualization approach to explain model predictions. Finally, two explainable AI tools and techniques named SHAP and Morris Sensitivity Analysis on the bidirectional LSTM regression model (discussed in section 5.2.2.3) are presented to better understand the model results in section 6.2.2.

This dissertation uses the words "interpretation"/ "interpretability" and "explanation"/ "explainability" interchangeably for simplicity. But many researchers suggest a distinction between these words [133]. For example, according to Miller *et al.*, "Explanation" is used to explain individuals or individual predictions, and interpretation should be used to justify an overall system, cohorts, or aggregated predictions [133].

6.1 Opening the Black Box: A Tool Allowing Advisors to Engage in Knowledge Discovery

This section will discuss a prototype of an interactive learning analytics tool. This tool is called "EAGER LEARNING ANALYTICS"[5]. This is the first work under this dissertation on introducing the idea of explainable AI to learning analytics research and how we can leverage explainable AI to understand a machine learning model better. The first published and implemented version of the tool is discussed in the appendix $A.2$ since the implemented version is a team effort. Other Ph.D. students implement the majority of the features of that version.

6.1.1 Motivation

Although the primary goal of many learning analytics tools is to help educational leaders (the domain experts) predict students' outcomes and plan interventions, these tools do not always explain the "why" behind those expected outcomes. One of the goals of this dissertation is to help the domain experts better understand their students by exploring the reasoning behind these predictions by getting involved with building the analytical models themselves. The tool attempts to do this by engaging them in the life cycle of analytics by providing the flexibility to select features to build better analytical models for predicting student success. This dissertation also aims to incorporate explainable machine learning to create helpful insights and intervention methods by opening the black box model for analytics. Based on these ideas, a prototype has been developed to conduct pilot user studies on the usefulness of this tool. The preliminary findings provide encouraging feedback showing how the users can get directly involved with the analytics and interpretation of results.

This tool will allow domain experts to choose features from a set of provided features and investigate which features significantly impact a student's performance and behaviors. Moreover, it may help academic advisors better understand their students and the "why" behind the particular predictive results. This tool incorporates explainable machine learning with visualization of the features to better understand the components used for analysis.

The approach of this dissertation differs from the previously developed tools as follows:

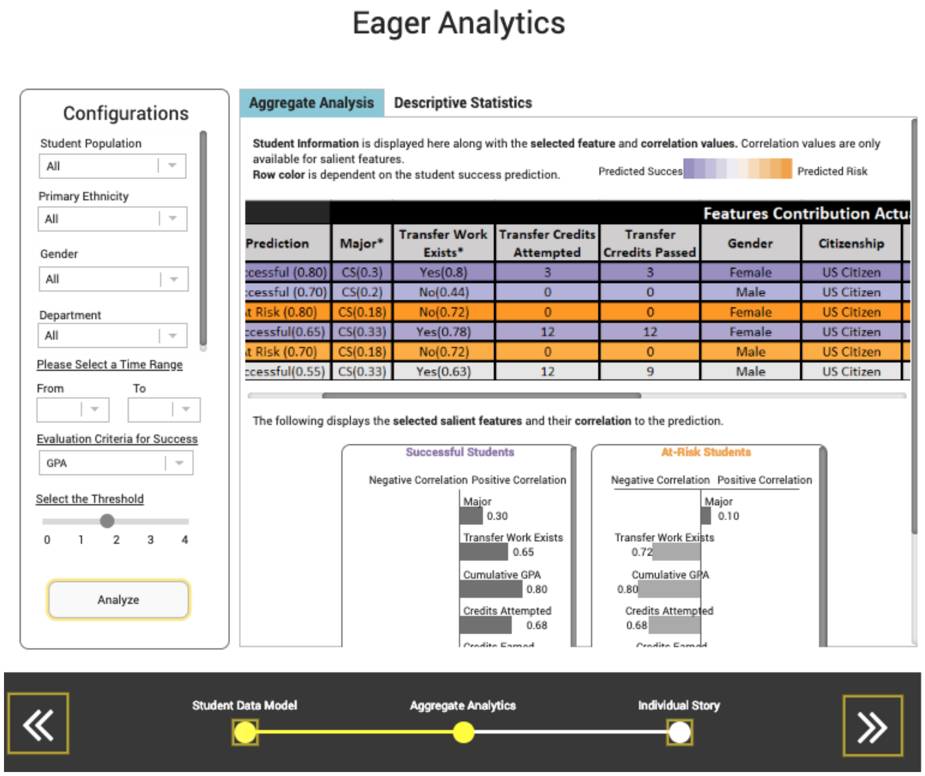
- The intention is to involve domain experts in the knowledge discovery lifecycle by adjusting the features that will be utilized for analytics to gain greater insight into the interesting, known, and unknown facts of the outputs.

- The domain experts are directed towards the data analysis by giving visual explanations of contributions of the features to a particular decision (success or failure).

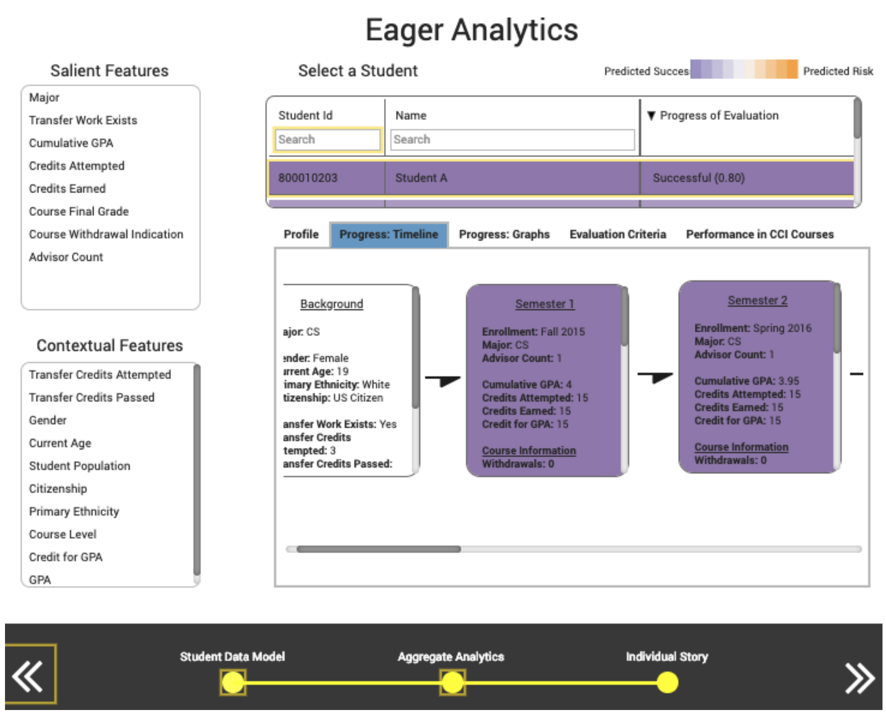
This dissertation addresses two main gaps in the existing tools. The first research gap is that most tools do not use temporal models for predicting students' at-risk [16, 17, 18]. However, the bottlenecks in solving this challenge is - 1) the heterogeneity of students' data hinders integrating different data sources for modeling and 2) the sparsity of some features that could be important for specific populations rather than others [3]. The second research gap addresses the "why" question that faculty leaders have in mind when analyzing students' data. Conati *et al.* [19] emphasize the need for explainable AI in open learner models, which shows the students the reasons behind their performance. Applying the same concept to LMS, the novelty of the prototype exhibits the importance of explainable AI for advisors to understand the model predictions.

6.1.2 Approach and Uniqueness

Paper prototypes were used first to brainstorm on the tool and then used "Indigo Design Studio" to build the prototype tool. This chapter discusses the prototype first. Next, it reviews unique factors of the prototype, including (1) the feature selection process, allowing domain experts to get directly involved with the model building, and (2) the use of explainable artificial intelligence.



(a) The Aggregate Analysis View



(b) Individual Story View

Figure 6.1: Prototype:Eager Analytics.

6.1.2.1 Prototype

The prototype mainly consists of three main views: (1) the student data model, (2) aggregated analytics, and (3) individual story.

6.1.2.1.1 Student Data Model

This view allows the user to select the features they wish to include in their model. These features are of three categories: (1) the background node, (2) semester nodes, and (3) the outcome node. The *background* node consists of the students' background information, such as demographics.The *semester* node consists of different features per semester, such as course information. The *outcome* node includes features for their performance evaluation, such as GPA.

6.1.2.1.2 Aggregated Analytics

This view first allows the user to configure additional options before finalizing the model, such as defining the criteria for student success. Once the model is created, information regarding the student cohort is presented. That information includes the predicted outcomes (e.g., percentages of successful versus at-risk students in the given cohort) and the correlation of various features to these outcomes (see Figure 6.1 (a)).

6.1.2.1.3 Individual Story

This view allows the user to select and explore an individual student, mainly by displaying the student's progress over time. This includes visuals on the student's background and semester-wise information (see Figure 6.1 (b)). A similar analysis is displayed as the aggregate analytics view, which describes the correlation of features for the individual student.

6.1.2.2 Unique Aspects of Our Tool

We incorporate explainable AI to create a tool to help advisors understand their students and the "why" behind these success/at-risk predictions. Rather than simply giving the predicted outcome of a student, our tool involves the users (e.g., advisors) in the analytics-knowledge discovery loop itself. This helps open up the "black-box" by allowing them to use their existing domain knowledge by flexibly selecting the features provided in the tool.

Furthermore, after selecting features, we use explainable AI to help them understand the importance of those features in predicting the outcome. To do this, we display the correlations of the features to the outcome (called explanations). For this prototype, we specifically use LIME [12] to generate the aggregated and individual correlations.

6.1.3 Results and Contributions

While building this prototype, we wanted to evaluate both the usefulness of the tool and the performance of the analytical models themselves. To evaluate the current progress of our prototype, we performed preliminary pilot user studies. When developing and testing our models, we focus on undergraduate students who have declared a major in computers at some time throughout their academic careers. As a result, we conduct our research for various periods depending on the data availability (i.e., last ten years, previous thirteen years, etc.). On-time graduation was chosen as the success criterion for the pilot study. A Neural Network (RNN) was implemented as our predictive model. We achieve an accuracy of 95% for a fixed length of semesters and a limited set of features. We want to verify the outcomes for eXplainable AI analysis for this prototype. Therefore, we will not go deeper into this RNN model.

Furthermore, LIME [12] was used to explain the RNN model to get insightful explanations. For example, LIME presents the correlation values of how previous high school rank and GPA per semester influence future semesters' performance.

6.1.3.1 Pilot Study Design

A pilot study with two individual faculty advisers was conducted to learn what users find important in a tool to support advising and if they think the explainable AI makes any sense. We recruited three faculty advisors who are already familiar with advising students and using different tools to advise. Firstly, we demonstrated the prototype with a seven-minute training video in the interview session with each adviser. Then, they were asked to explore the tool with a fixed set of four tasks. Each task also contains some subtasks. One of us observed their interactions with the tool and took notes. Finally, they were asked some reflection questions after finishing each of the tasks. The first three tasks consisted of exploring the individual modules of the tool - 1)the student data model, 2) aggregate analytics and 3) individual storytelling. In the fourth task, we asked them to edit different features in the student data model and observe the changes. We finished our pilot study by asking them about the tasks, individual features, modules, and how they feel about our technology.

6.1.3.2 Feedback from the Pilot Study

We receive some valuable feedback from the two participants in this user study :

- Overall the student data model page was easy to understand and explore. The instructions were informative, and users were comfortable exploring the various features. They also liked that we included "veteran information" and "primary ethnicity" as the available information.

- By allowing users to define the threshold and evaluation criteria for success, more flexibility is added. It will enable users to understand better what factors are going into the predictions.

- Users thought that the progressing storyline in the "individual story" page could help them design interventions for their students.

- Both of them liked the explanations with feature importance provided by LIME in the "aggregate analytics." Using gradient colors to understand the positive or negative effect of a feature on the model outcome was helpful. But they asked us to add more features and more training on it.

- Both of them thought that we should have invited some academic advisers, not only the instructors. They felt that academic advisers have more knowledge on the features we should use.

- In the study, they also provided us with some guidelines on the tool design. For example, adding color-blind-friendly color pallets, filtering, sorting, proper naming, labeling, and searching capability would make the tool more acceptable.

In summary, we used a human-centered design approach to engage faculty and advisors in developing an interactive knowledge discovery tool to understand student success and students at risk better. Our method allows faculty and advisors to interact with the data used in an analytic model, the results of the analysis, and the story of individual students.

Section 6.2 will demonstrate the usefulness of several eXplainable AI (XAI) tools and approaches on two types of recurrent neural networks (discussed in section 5.2.2).

6.2 Interpretation of a Black Box Model

Interpreting a neural network model is challenging since the model is a black box. The data input of a primary neural network passes through many layers of non-linear transformation and multiplication of learned weights to make a single prediction. Humans are unable to follow this complex mapping from input to output prediction. It is impossible to perceive the complex interaction between the millions of weights within a neural network. To interpret the behavior, concepts, and forecasts of neural networks, we need interpretation methods.

In section 6.2.1, we discuss a state-of-the-art XAI tool for classification analysis using the vanilla LSTM RNN. Section 6.2.2 includes a discussion on the deep neural network for regression analysis (Bidirectional LSTM RNN) with some eXplainable AI(XAI) tools and techniques.

6.2.1 SHAP: Analysis of Prediction Results of Vanilla LSTM RNN Classification Model on Student Data

This section will discuss the feature importance and predictive results analysis on LSTM RNN classification model with SHAP explainable AI tool.

SHAP (SHapley Additive exPlanations) [132], a game-theoretic approach to understanding the prediction output of any machine learning model, will be discussed in this section. It uses the SHapley values from game theory and related extensions to combine optimum credit allocation with local explanations [134, 73].

Here, we describe the local explanations of the RNN model with SHAP. The authors integrated many prior works on local illustrations with the concepts of Shapley values into this tool. Since we are describing a sequential neural network model with the tool, we will be using the DeepExplainer algorithm, which computes SHAP values with the combination of the concepts from DeepLIFT [135]. DeepLIFT presents a framework for learning the critical features in a deep neural network by propagating the deep layers' activation differences.

DeepExplainer algorithms cannot handle explanations with categorical data in sequence models. We will show different plot representations that may help the data science practitioner understand the importance and verify if the features are working as expected.

SHAP (SHapley Additive exPlanations) is a unifying strategy for explaining any machine learning model's output. As presented well on the *GitHub* page, SHAP connects game theory with local explanations. SHAP can use $3D$ data as an input, unlike other black-box machine learning explainers in Python. The output in this scenario will be $3D$ data with essential values for $[eachsample, timesteps, features]$ in that sequence. We get $2D$ data with feature importance as columns and time steps as rows if we sum the output on the $0th$ dimension (i.e., sample axis). SHAP prioritizes time-stepwise features in this approach. We have taken 500 random samples from the training dataset as a test subset to pass in DeepExplainer class as this much data shows very accurate results. The difficulty of this method rises linearly with the number of data points. Therefore supplying the complete train data will produce precise high values but will be prohibitively expensive.

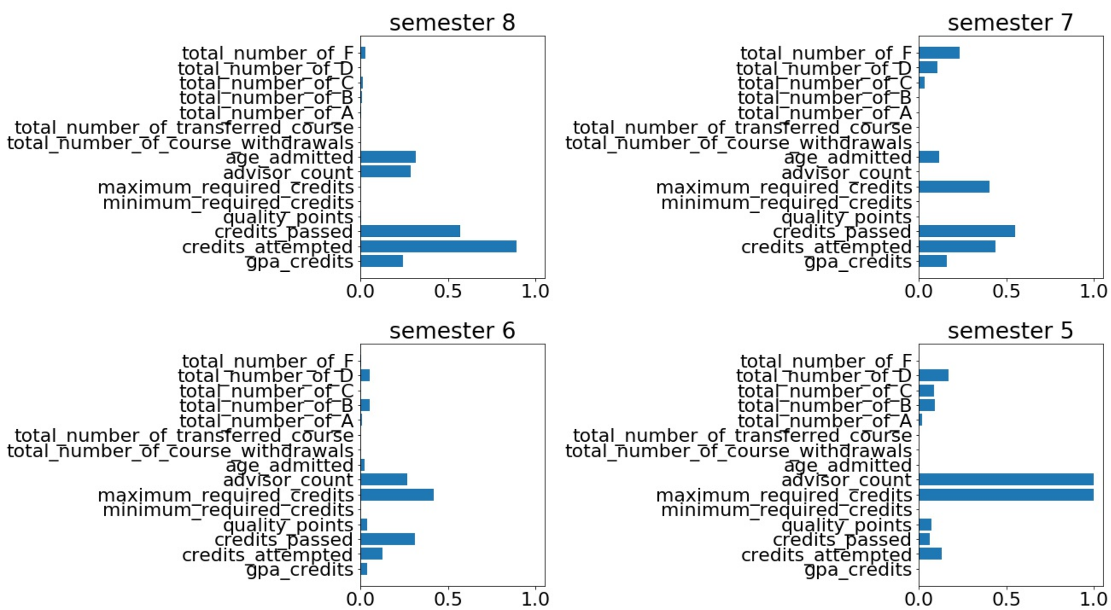


Figure 6.2: Shap values for fifth to eighth semesters of data on a random sample of 500 students

Figures 6.2 and 6.3 are showing feature importance on eight time-steps (eight semesters). To predict class labels, we looked at data segments with only eight semesters of data. We can make a conventional bar plot by taking the mean absolute value of each feature's SHAP values (produces stacked bars for multi-class outputs). The values displayed here are the output of shap values summed up on the sample axis. We can explain it step by step with each semester. With this explanation, we can involve domain experts to understand the model's efficacy by matching their advising or policy-making experience. The following eight subplots show how

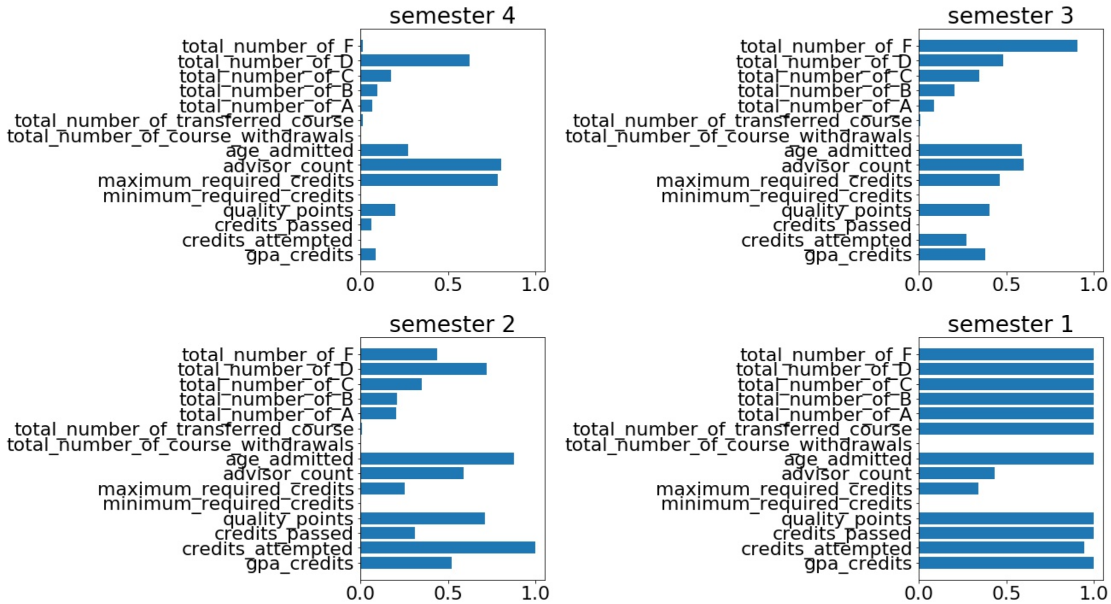


Figure 6.3: Shap values for first four semesters of data on a random sample of 500 students

the feature importance (SHAP values) changes over semesters.

Even though near future (8th semester) values matter more in success and risk prediction decisions, these plots show that the more immediate time steps' features with higher shap values also demonstrate how the change in shap values occurs from the beginning to the nearer semester's features.

To identify which characteristics are most important for a model, we can plot the SHAP values of each feature for each sample. The total of SHAP value magnitudes over all samples is used to order features, and SHAP values are used to depict the distribution of the effects each feature has on the model output. The color represents the value of the characteristic (red high, blue low). Figure 6.4 shows a summary plot on how the shap values of all the features reflect on the model. It offers the lower value of each feature (since there are no red dots) and has zero positive and negative impacts on the model. It is not showing how the higher values (no red dots) of the feature will impact the model. This kind of explanation is helping us to decide that the model needs more careful attention while developing it. But it is still helpful in determining how we should design the explanations of the model. We need to see the effect of features on different cohorts of students. Because if we see the gpa_credits, the lower values of GPA credits positively and negatively impact the model, making no sense.

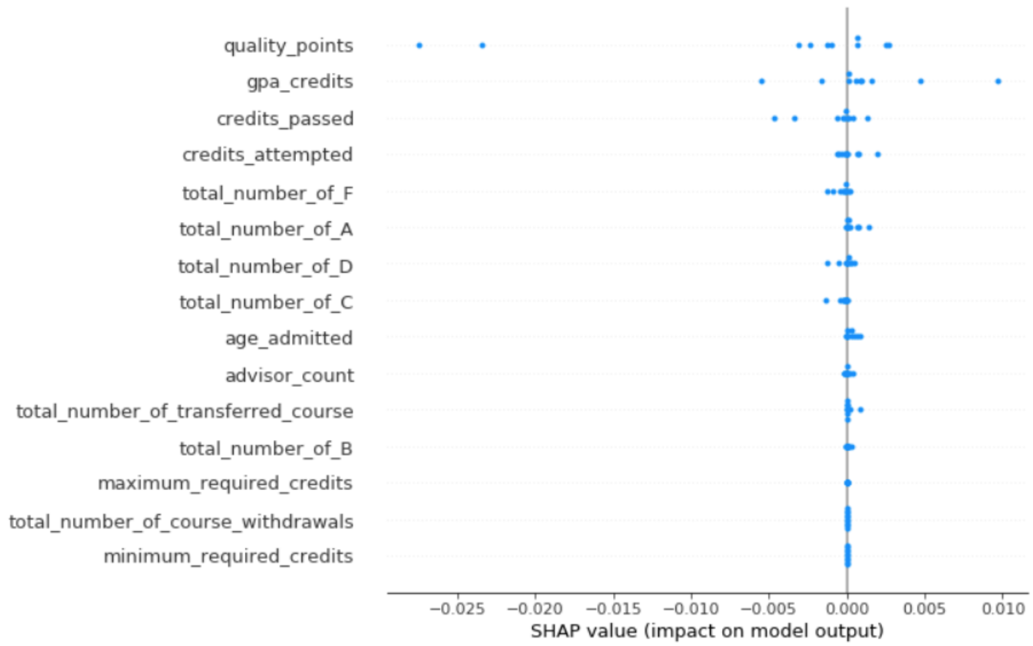


Figure 6.4: Distribution of aggregated SHAP values for the features on the whole student data

The following force plots are the same as summary plots. But it aggregates the effect of features on each cohort, i.e., the cohort of at-risk students or successful students. The red color means the higher value of features, and the blue color indicates the lower value of the features. We will be describing our understanding along with the images of the force plots. The force plots show - 1) what kind of effect the feature has on the cohort by highlighting the color with red or blue and 2) the magnitude of the effect and the relative value of the feature. For example, with a low magnitude of blue highlight for credits attempted, a lower number of credits were attempted, which has a low effect on that particular cohort and the model output.

In figure 6.5, the aggregated value of only quality points on the cohort of successful

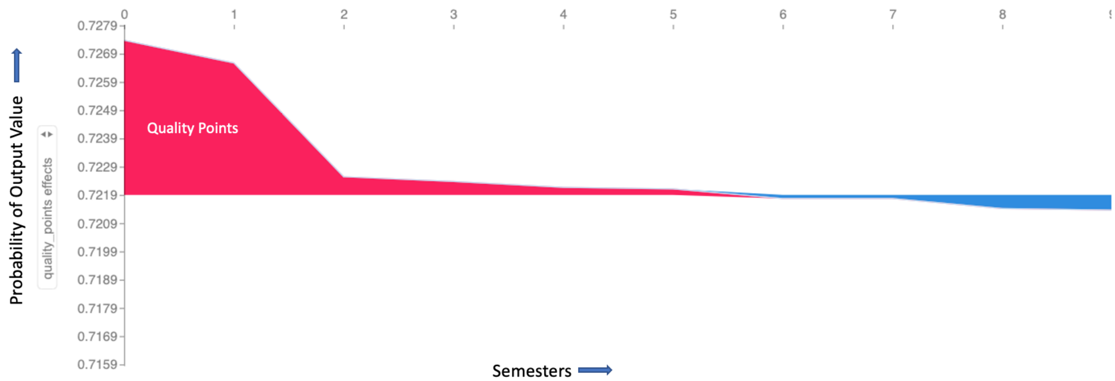


Figure 6.5: Effect of quality points on the successful studentsâ cohort

students shows how at the beginning (at least until the $5th$ semester), high quality points significantly impact success. At $5th$ and $6th$ semesters, it has both lower and higher effects meaning that the effect is distributed among the successful students. In the later semesters, quality points have a softer impact with a lower magnitude on the prediction.

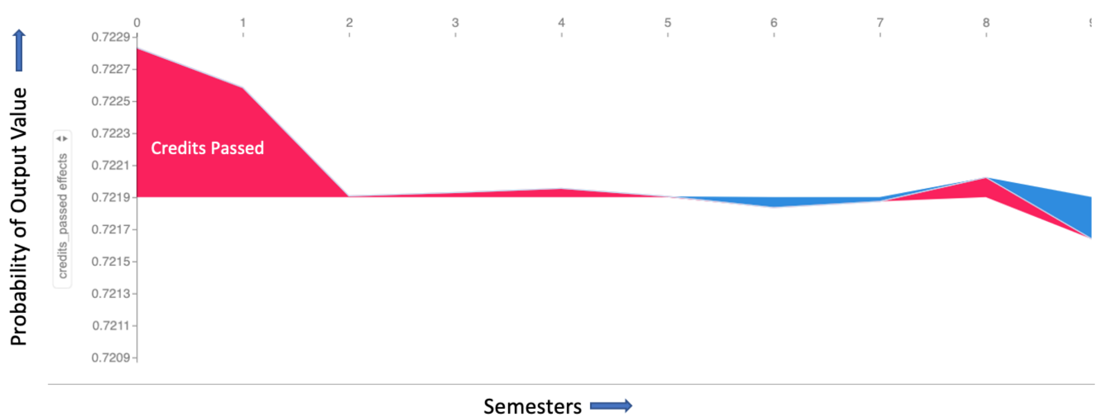


Figure 6.6: Effect of credits passed on the successful studentsâ cohort

In figure 6.6, credit passed also aligns with the effect as the quality point on the successful students' cohort like Figure 6.5. It also proves how quality points are positively correlated with the credits passed.

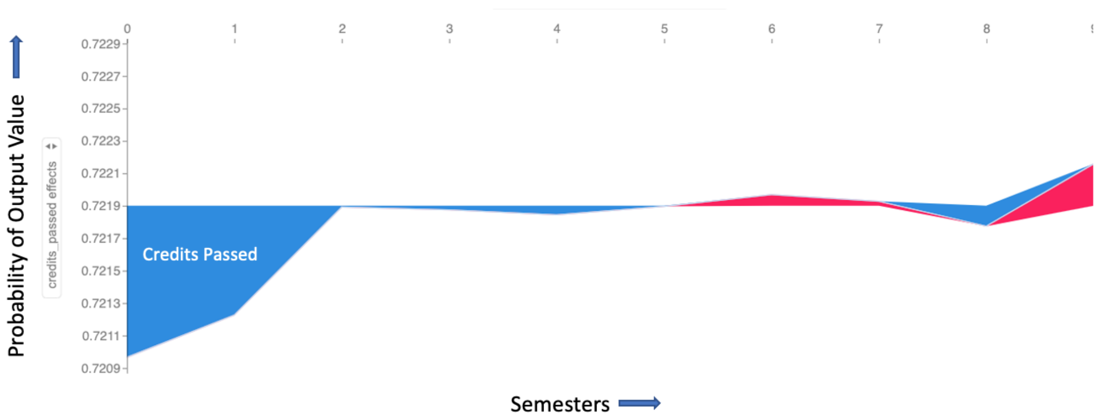


Figure 6.7: Effect of credits passed in at-risk studentsâ cohort

In figure 6.7, we can see that the effect of credit passed is precisely opposite to the cohort of at-risk students. The lower values of credits passed contribute to the lower values of CGPA, therefore, contributing to at-risk students.

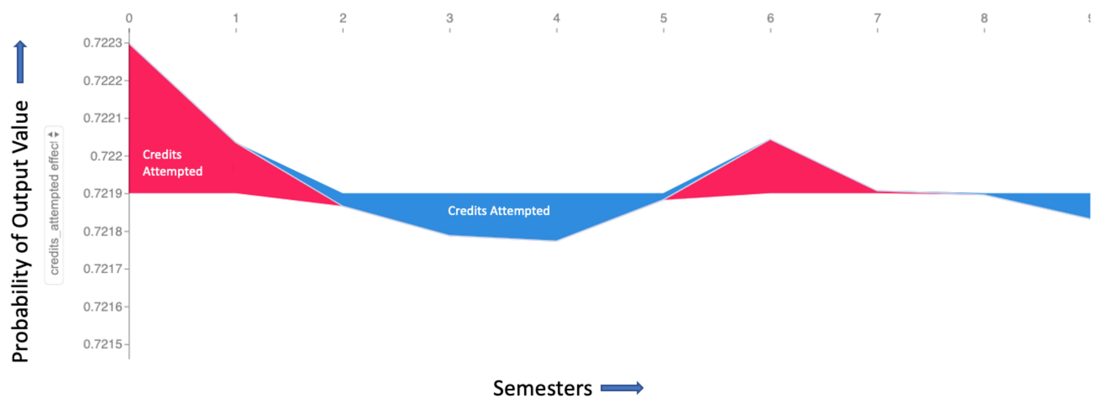


Figure 6.8: Effect of credits attempted in at-risk studentsâ cohort

Figure 6.8 represents the effect of credits attempted for the first nine semesters on students' at-risk cohort. If we combine the analysis in Figure 6.7 and Figure 6.8, we will see that at-risk students are used to taking a higher number of credits at the beginning. But since the number of credits passed is lower in the middle, i.e., from $3rd$ to $5th$-semester, at-risk students will attempt a lower number of credits and a lower number of credits passed, but both have a softer impact on model prediction. Finally, from the $5th$ to $7th$ semester, the at-risk students try to improve their conditions, so they attempt a higher number of credits and eventually, with fewer credits passed. In this time range, both the credits passed and attempted features have a higher impact on model prediction. According to both the plots, the behavioral pattern of attempting higher credits and passing credits positively affects the model to predict the pattern of at-risk students.

6.2.1.1 Summary of Findings

1. The SHAP explanations demonstrate how the features act on the model, but it also presents the typical feature patterns for a particular cohort. For example, Figures 6.5 through 6.8 showed how quality points, credits attempted, and credits passed are behaving differently for different cohorts (successful and at-risks). It will also help the decision-maker or the model builder decide if the model is acting logically (which it is) and if it needs improvement. For example, Figures 6.5 through 6.8 demonstrate that the model is doing an excellent job of extracting the student cohorts' patterns. But figure 6.4 is not that helpful since it explained the features with lower impact on the model, which cannot be accurate.

2. We learned from figure 6.4 about what we should expect from explanations and where explanations might be helpful. After seeing the explanations of figure 6.4, we understand that identifying feature importance on the whole student data might not be intuitive since the effects of the features are supposed to be reversed for two different cohorts. And it leads us to leverage the explanations in figures 6.5 to 6.8.

3. Besides showing the feature importance for particular cohorts, SHAP explanations explain how different features play an essential role in the time series analysis (see Figures 6.2 and 6.3). This kind of explanation is also beneficial to plan early interventions.

4. SHAP analysis shows us the exact scenario we want to see. Yet, our vanilla LSTM classification model is not doing an excellent job since the data is imbalanced, and the model predicts the majority class. Domain experts in our research group suspect that the highly correlated features may play a vital role in the SHAP explanations. We change our analysis to a regression model. We will see how the SHAP analysis can help us analyze the regression problem in the next section.

5. SHAP summary plot analysis demonstrates how the behavior of a model changes over time in terms of feature importance. This will help the data scientists to decide which features are relevant for building a model and therefore, refining an existing model.

Since we are determined to involve the non-data scientist decision-makers or domain experts in learning the model's efficacy and credibility, SHAP will be the better fit as the explainable AI technique. Moreover,SHAP will also help the data scientists in the feature selection process.

6.2.2 Interpretation of A Regression Analysis with Recurrent Neural Network

This section will discuss the SHAP analysis on a regression model to understand better what features are relevant to our chosen Bi-directional LSTM model. We discuss Morris Sensitivity Analysis which ranks the features by how much they can derive the model output variability [136].

6.2.2.1 SHAP Explanations on the Regression Analysis of Bi-directional LSTM RNN Model

In this regression model, we discard the features which are directly correlated to calculate the CGPA. Instead of those features, we use some engineered normalized features, for example, percentage of A, course progression, etc.

Since it is regression analysis, we cannot demonstrate the force plots (like Figures 6.5 to 6.8) for two different student cohorts (successful vs. at risk). But we think the force plots are still helpful to understand the features' importance throughout a student's first eight semesters. For example, as we can conclude by figures 6.2 to 6.4, the "minimum required credits" do not play any role in the model prediction. This observation is persistent for the regression model too. The force plots for âminimum required creditsâ and "GPA Type" show no effect on any force plots.

We will now discuss force plots of some of the features as follows.

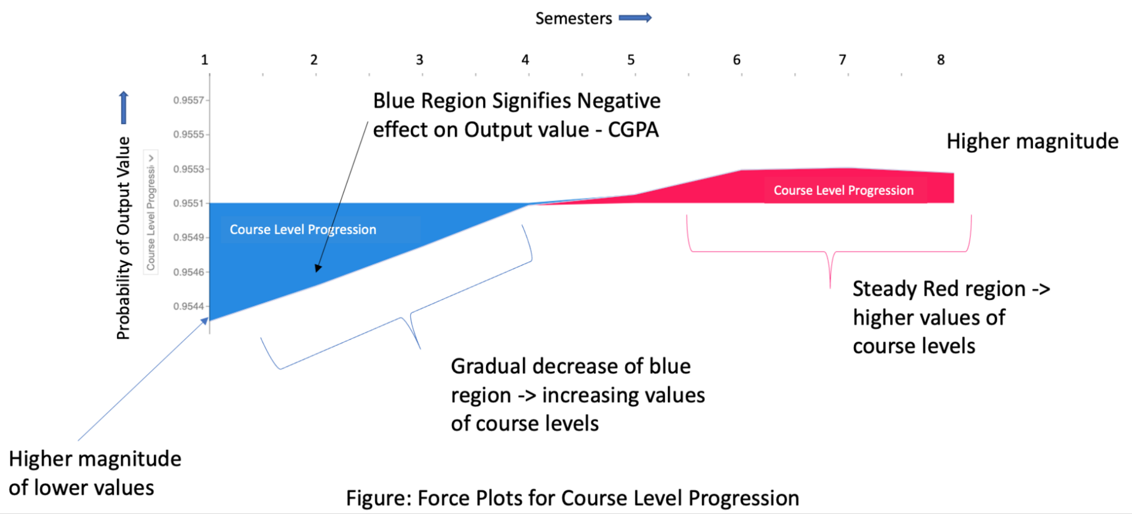


Figure 6.9: Effect of "Course Level Progression" on the value of CGPA

Figure 6.9 shows an example of a SHAP force plot on the course level progression feature. The x-axis represents the time for the analysis. We sampled the students with at least eight semesters of data. Therefore, the x-axis shows the interaction of course level progressions with the outcome CGPA through eight semesters. The y-axis represents the probability of output value, where the value 0.9551 represents the base value where the x-axis has started. It means 95% of the time course level progression will demonstrate this pattern with the prediction output.

We see two colors in this plot - one is red, and another is blue. Red means the feature has a positive effect on the higher value of the model output, and blue has a negative impact on it.The magnitude of the blue region also has significance to this visualization. A higher value of the blue region means at the beginning of the first semester. The higher magnitude of lower values of course progression is correlated to having a higher value of CGPA. Our unsupervised clustering analysis shows that course-level progression has a high correlation with the students' performance. We see the clusters having the higher number of successful students show a steady increase of course level progression. The force plot also supports our hypothesis. It suggests that a steady rise from lower(with the blue region in the earlier semesters) to higher values (with the red area in the later semesters) contributes to the higher value of CGPA.

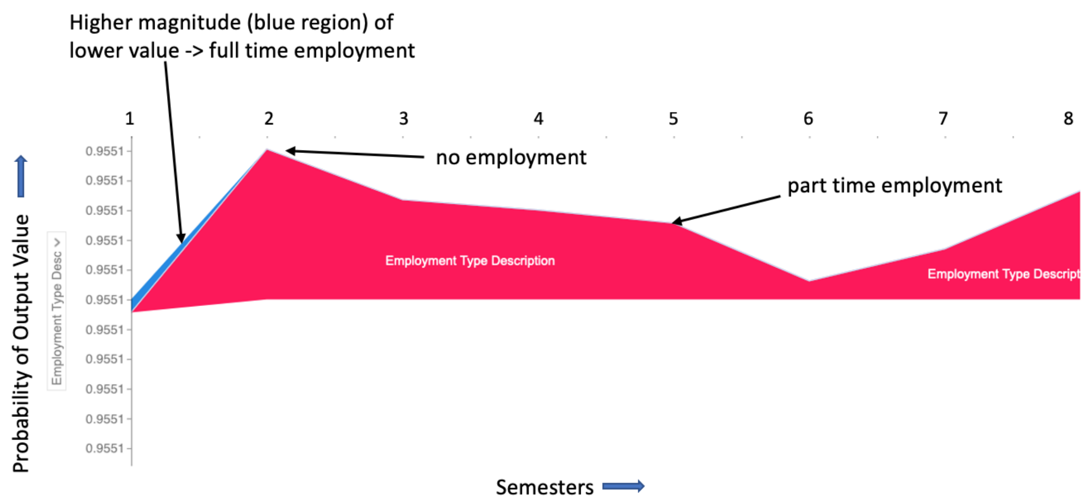


Figure 6.10: Effect of "Employment Type" on the value of CGPA

We use one-hot encoding for the categorical values such as *employmenttype*. There are three encoded values for this feature - 0 (which is a lower value) represents full time employed, 1 means *partiallyemployed*, and 2(which is a higher value) illustrates no employment. From the force plot in Figure 6.10, it is clear that with higher values (partial or no work) of the employment type, students are more prone to perform better since they have fewer responsibilities and more time to study. We see a small blue region at the beginning coexisting with the red area, which means having or not having employment at the beginning of the undergraduate career can result in a good or higher CGPA since at the beginning curriculum is much easier.

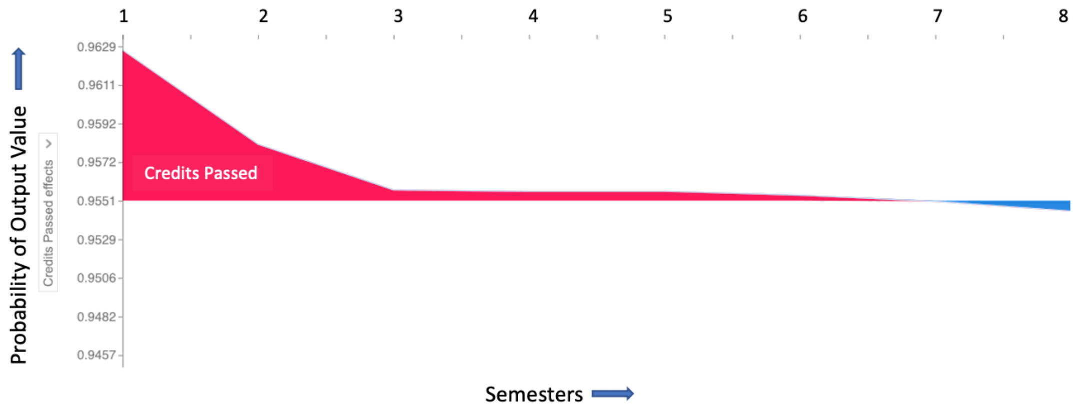


Figure 6.11: Effect of "Credits Passed" on the value of CGPA

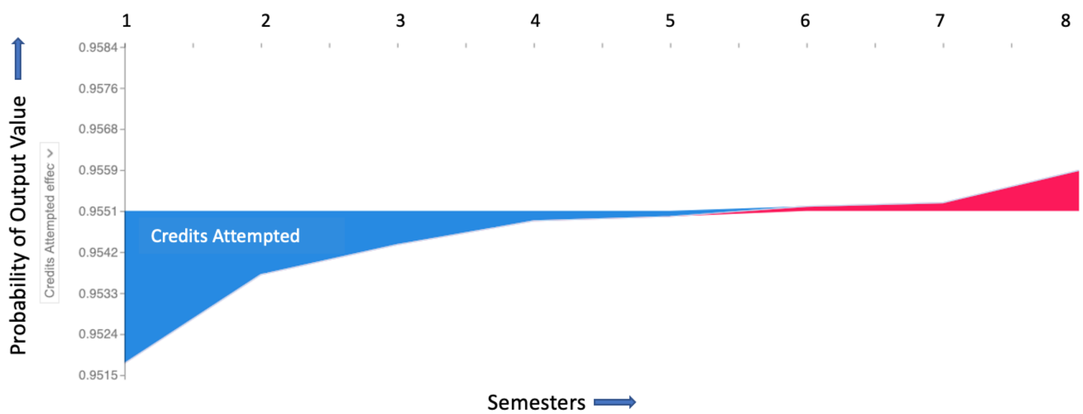


Figure 6.12: Effect of "Credits Attempted" on the value of CGPA

We see similar effects of credits passed (see figure 6.11) and attempted (see figure 6.12) in the force plots as our earlier LSTM classification model. A lower value of credits attempted helps the students to achieve a higher value of credits passed, and that is how they are both related to achieving a higher CGPA at the beginning of their career. Around the last semesters, it is required to attempt more credits to earn a better CGPA if the value of credits passed gets slightly lower.

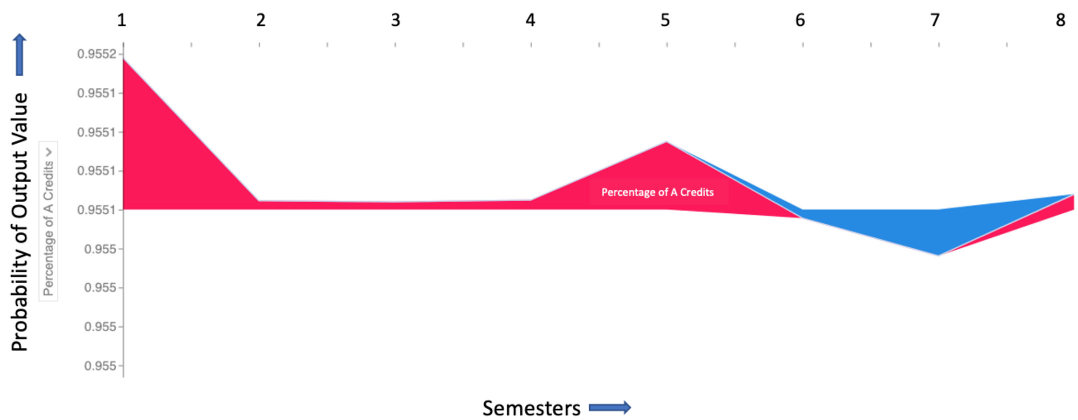


Figure 6.13: Effect of âPercentage of A creditsâ on the value of CGPA

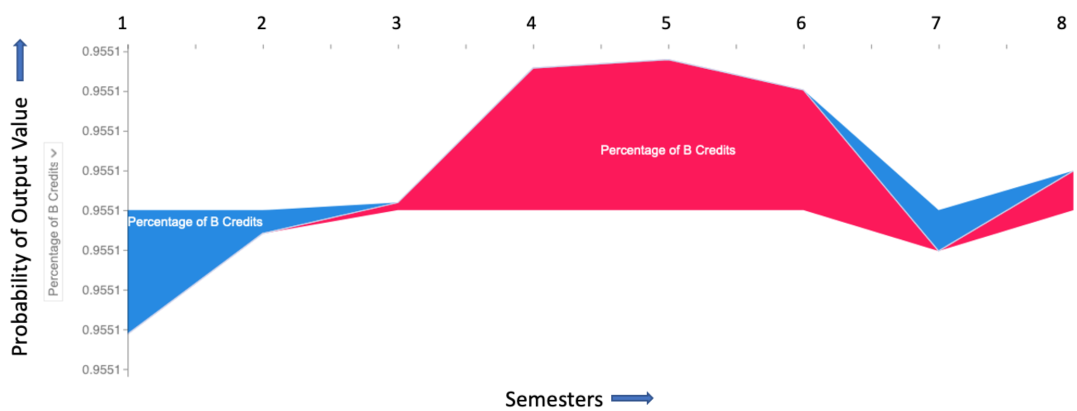


Figure 6.14: Effect of "Percentage of B credits" on the value of CGPA

The force plot(see Figure 6.13) of the percentage of A credits is intuitive. Higher values of the percentage of A credits significantly affect the higher CGPAs at the beginning semesters. However, at the finishing semesters, both the blue and red regions coexist together, which means that still, the students can have a higher CGPA at this time, if at the beginning, they maintain a good percentage of A credits. Percentage of C, D, and F credits have the exact opposite patterns. We include the force plots to demonstrate the effect of percentages of C, D, and F in appendix A.3. However, we see a mixed pattern in the percentage of B credits (see Figure 6.14).

Figures 6.15 and 6.16 show a change of feature importance on eight time-steps (8 semesters) like Figures 6.2 and 6.3 of the classification model. Same as the force plots, GPA type and minimum required credits play no roles in forecasting throughout the eight semesters. In the beginning, many of the features have higher importance. But over time, their effects diminish. All other features, especially employment type and course progression, present the same pattern discussed in the force plots. Therefore, this alternative form of explanation and visualization may help understand the correlation between students' behavior and performance over time. Furthermore, it allows us to exclude elements that aren't necessary for the model to learn.

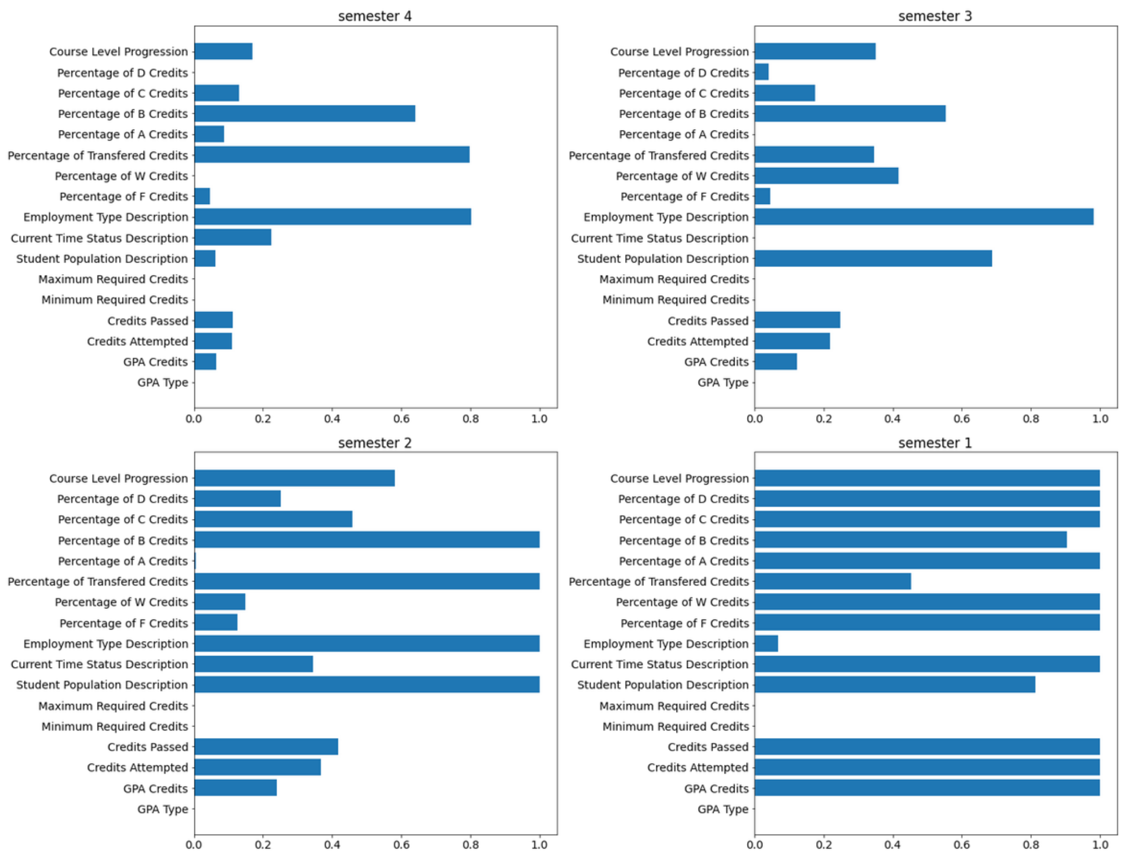


Figure 6.15: Shap values for first to fourth semesters of data on a random sample of 500 students

6.2.2.2 Morris Sensitivity Analysis on the Regression Model

A combination of methodologies known as sensitivity analysis captures how a model reacts to changes in its inputs. We use these statistical methods to discover how variation propagates from inputs to outputs [136]. Saltelli *et al.* define sensitivity analysis as describing the "relative relevance of each input in determining the output variability." Therefore, the result of such algorithms is a ranking of inputs based on their sensitivity. Sensitivity analysis can be carried out at both a global and a local level. Local approaches focus on input variation around a specific point in the input space. In contrast, global methods provide summary statistics of the effects of input variation across the entire input space.

For global sensitivity analysis, the Morris technique is used as a statistical method. This method was designed as a preliminary computational experiment to grasp the relative influence of each component and is of the "one-factor-at-a-time" type. The Morris method assigns an individual element to the distribution of elementary effects (EE). A mean $Mu(\mu)$ and a standard deviation $sigma$ (σ) exist for each EE distribution. These two statistics are what allows the features to be mapped into several classes. When the model is non-monotonic, the mean can be negative. Hence a Morris method variation accounts for this with absolute values $Mu^*(\mu^*)$ to make it easier to read. We'll use the variation in our analysis.

Morris sensitivity indices, such as the mean(μ) and standard deviation elementary effect (σ), as well as the absolute value of the mean(μ^*), are calculated for our Bidirectional LSTM regression model. It's easier to visualize these numbers in a tabular format to sort and color-code them. The factor's overall importance is denoted by μ^*. On the other hand, σ demonstrates how much a feature interacts with other features. μ^* is the mean of the output variation given the variation of each input, sigma (σ) is the standard deviation of that variation in the output. Therefore, the Morris analysis gives us insights on the most critical parameter; credits attempted in this case. But it also tells us which parameters interact with other parameters, and that is given by sigma. So credits attempted is the most crucial feature, but it also has interactions with other parameters, which is why it is non-linear.

For our analysis, we group the normalized GPA credits (i.e., Percentage of A, B, C,

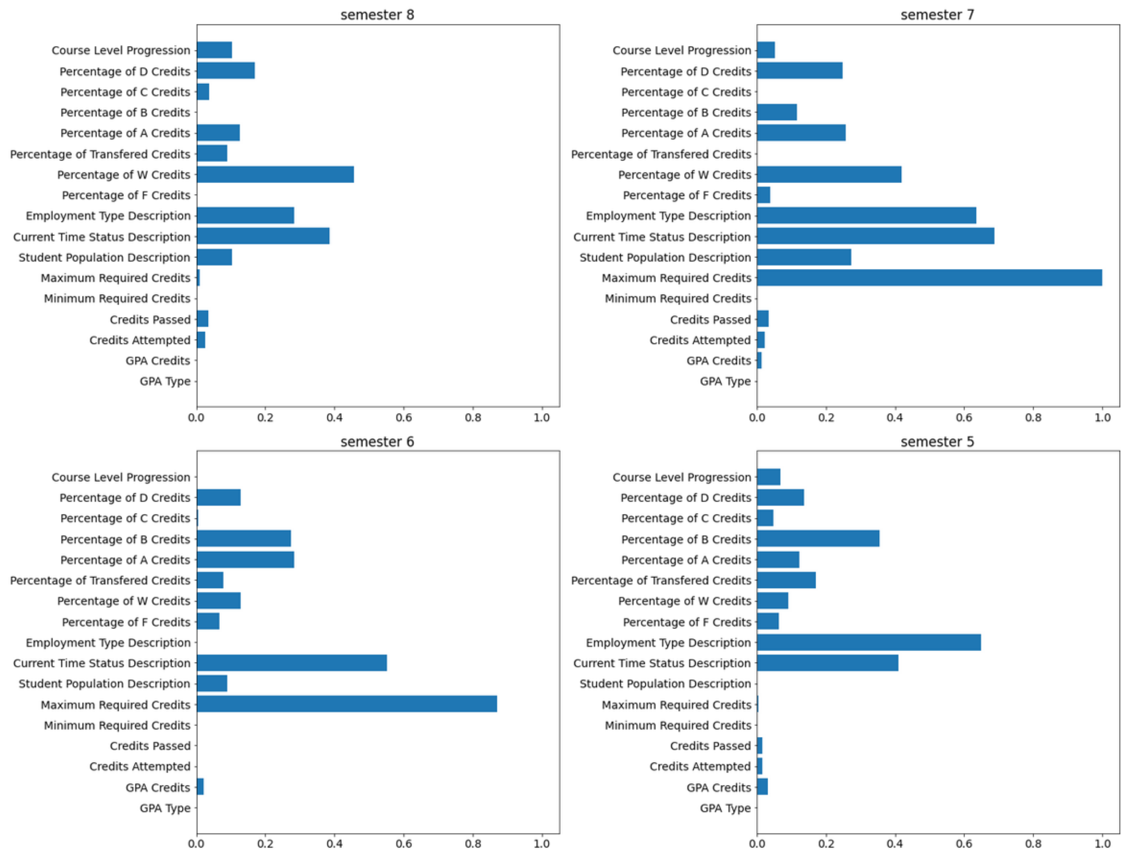


Figure 6.16: Shap values for fifth to eighth semesters of data on a random sample of 500 students

D, F) into the feature "Normalized GPA" (see Figure 6.17). It's worth noting that the "Normalized GPA" grouped feature does not have an absolute mean elementary effect but an inconclusive interaction effect. Groups are challenging to assess, especially when they are very sparse. We see the feature credits attempted have the overall importance influence signified by a highest μ^*. A higher σ that credits attempted is highly involved in interaction with other features. We do not report the other features in the figure 6.17 since they have very insignificant values.

features	μ	μ*	σ
Credits Attempted	-422.244843	1473.238647	1562.508057
Normalized GPA	nan	374.364299	nan
GPA Credits	130.370255	335.439972	550.448364
Current Time Status Description	64.467247	203.200897	415.329010
Course Level Progression	-31.056435	61.469185	154.416183
Employment Type Description	-3.240711	27.507318	111.302834
Percentage of Transfered Credits	13.321960	19.999458	75.782249

Figure 6.17: The Elementary Effects (EE) decomposition of the features

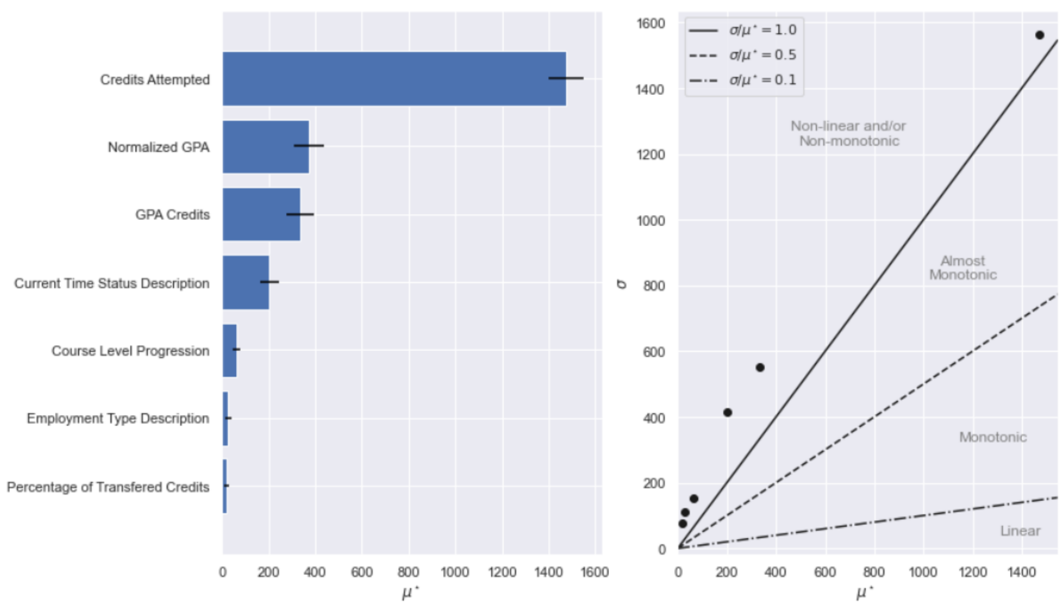


Figure 6.18: A bar and covariance plot depicting the Elementary Effects (EE)

In Figure 6.18, the bar plot on the left panel ranks the factor by mean μ^*, while the line sticking out of each bar represents their corresponding confidence bands. The scatter figure to the right shows the covariance with μ^* on the $x - axis$ and σ on the $y - axis$. As a result, the further right the point is in the plot, the more crucial it is, but the higher it is in the plot, the more it interacts with other aspects and becomes less monotonous.

Naturally, this means that features that do not interact much and mostly monotonic comply with linear regression assumptions, such as linearity and multicollinearity. However, the spectrum between linear and non-linear or non-monotonic is determined by the ratio between μ^* andσ. From figure 6.18, we can tell by the preceding covariance plot that only credits attempted are non-linear or non-monotonic. "Credits attempted" is the most important, with the following two (normalized GPA and GPA credits) clustered around almost monotonic areas. All the other features fall into the linear region.

6.2.2.3 Summary of Findings

Here is a summary of the limitations and findings from the analysis and observations of the explainable AI tools on the regression model.

1. Like the SHAP explanations on the classification model, SHAP explanations on the regression also demonstrate how the features act on the model and reveal the typical patterns of a higher CGPA value in the regression model.

2. Since we are working on a regression model, we cannot analyze the force plots on different students' cohorts. However, we do not face any problems with this. The force and summary plots are very intuitive. They demonstrate the segregated patterns for each feature from first to finishing semesters and how the patterns relate to the values of CGPA over time.

3. We can even interpret the results for some categorical variables such as "Employment type" and how these variables impact the students' performance by the SHAP explanation.

4. In the Morris sensitivity analysis, elementary effects help us understand how to classify our features following their impact on model outcomes. However, the visualizations are less intuitive to quantify their impacts on feature interactions properly. We need an analysis that can decompose the output's variance and trace them back to the inputs.

5. In Morris sensitivity analysis, most cases agree with SHAP in terms of finding top or irrelevant features.
   (a) Employment type description, Course level Progression - both XAI tools agree that they are top features
   (b) GPA type, Minimum required credits - both XAI tools agree that they are irrelevant features
6. The order of the feature importance may not agree with each other.
   (a) SHAP summary plot orders the features as-
      i. Employment type description $>$ Percentage of transferred Credits$>$ Normalized GPA credits $>$ Current time status description $>$ Course Level Progression
   (b) Morris Sensitivity analysis orders the features as-
      i. Credits Attempted $>$ Normalized GPA credits $>$ Current time status description $>$ Course Level Progression $>$ Employment type description

Morris analysis is a suitable XAI technique for data scientists to train their models. However, it is not ideal for domain experts. SHAP also presents the feature importance over time. Therefore, both SHAP and Morris sensitivity analysis can help the data scientist to extract the relevant features of the model, and to understand and refine the model better. For example, both the SHAP and Morris analysis methods show that "GPA type" and "Minimum required credits' ' are irrelevant features. We removed the features and reran our optimal bidirectional model on the other features. Table 6.1 reports the results of the prediction. It seems after removing those features; the performance improved a little bit. But still, the number of misclassifications is high.

Table 6.1: Prediction performance records with the Bi-directional LSTM after removing irrelevant features (GPA type and Minimum Required Credits) identified by both SHAP and Morris Analysis

# semesters	Accuracy (%)	Precision	Recall	F1 score	#Mis-classified	#missed in the class $0-2.00$	Total #students (Successful: At-risks)
1	92.39	0.88	0.92	0.891	492	220	6466
2	92.77	0.89	0.93	0.901	408	111	5646
3	91.76	0.8723	0.93	0.881	427	120	5188
4	92.96	0.844	0.935	0.899	325	93	4620
5	92.02	0.85	0.92	0.88	332	131	4163
6	92.79	0.8367	0.9237	0.89	236	65	3729
7	90.03	0.835	0.91	0.87	326	70	3278
8	92.58	0.834	0.913	0.87	201	33	2711
9	92.59	0.842	0.921	0.88	147	43	1993
10	91.93	0.83	0.919	0.873	115	57	1427

6.2.3 Insights Discovery

Figure 6.19 summarizes the three main building blocks of the LSTM RNNs analysis on the students' temporal features. We discover some findings on both the classification and regression models, and they lead us to some insights we can leverage from the model to plan intervention.

The main takeaways that can be drawn from these RNN models are:

1. Data imbalance is a significant problem for any machine learning model. We get poor uninterpretable performance results for the classification model. Moreover, to train a deep neural network, we need a high volume of data which may reduce imbalanced data for training on much larger datasets. Later, with the increase in batch sizes with cumulative data, we provided more data. However, that induces an increased imbalance of classes in the data.

2. Feature selection is one of the significant steps in training a model for better performance. While using the highly correlated features with output variables

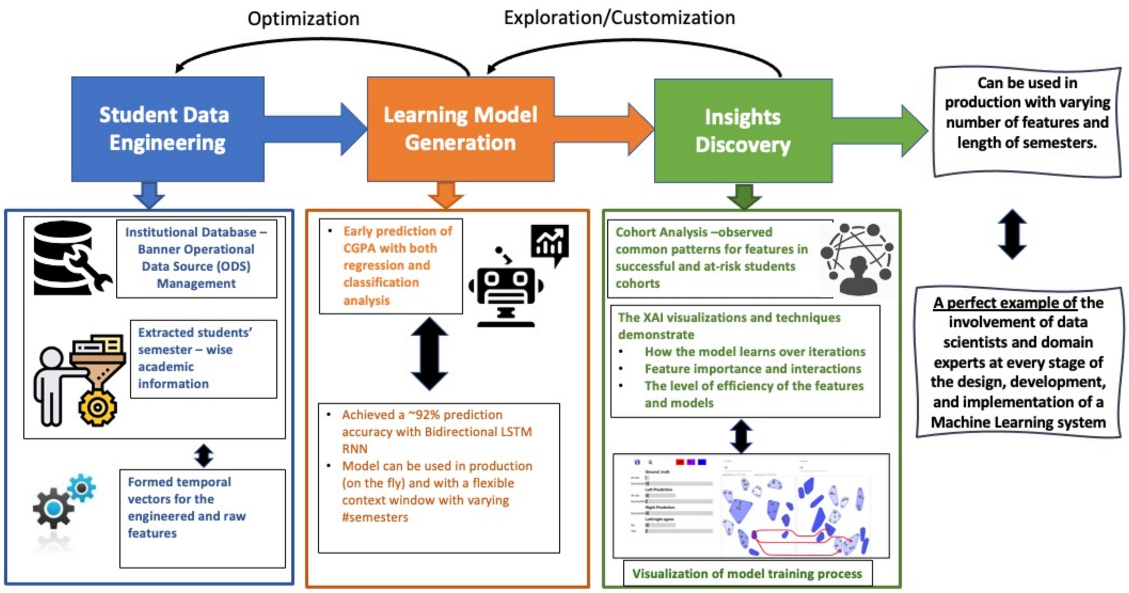


Figure 6.19: FIND on Supervised LSTM RNN on Temporal Data

lose credibility, using non-temporal data for temporal LSTM cells in RNN may corrupt the structure of the time-series model. However, we also have to be careful about any sampling methods. For example, oversampling increases the replicas in our data, leading to a higher number of false positives and negatives.

3. The time-consuming and rigorous hyperparameter tuning is ultimately beneficial for finding the best model for a dataset. Therefore, we achieve better performance and interpretability on our bidirectional LSTM regression model.

The findings of the RNN models and the explainable AI techniques assist us in extracting the following insights from the student data:

1. More data may help us understand the students' behavior and performance and the model's performance better. Even with interpretable black-box analysis, lesser volumes of data fail to describe the model performance and the predictive results.

2. SHAP explanations do a great job explaining the features of student perfor-

mance by cohort analysis, timewise projection analysis, and incorporating feature importance. Such explainable AI tools provide the insights and guidance which are helpful for the domain experts. The visualizations are very intuitive and graspable; with a minimal amount of training, non-data scientists will achieve a better understanding of the model's predictive results. This helps put trust in the machine learning systems. The non-data scientist domain experts can also improve these systems by incorporating their expert knowledge on students and student data.

3. Explanations provided by the Morris sensitivity analysis are helpful to train and implement the models. And the data scientists can get help with the relevant and essential feature selections. But these tools are not suitable for domain experts since they require expertise in data science and visualization to operate and utilize.

6.2.4 Lessons Learned and Limitations

The significant limitations and takeaways from these supervised learning models are three folds:

1. Describing the model's performance and demonstrating its credibility requires a high volume of data for the neural network models.

2. The data imbalance makes the analysis challenging to understand the model's performance and efficacy.

3. There is still a problem with interpreting results for some features, even with the XAI tools.

The involvement of the eXplainable AI (XAI) tool in the feature selection process improves the model's accuracy. But the visualizations and interpretations by the XAI tools do not show any significant patterns. Examples of such features are GPA type, GPA credits, percentage of transferred credits, etc. Finally, finding the most likely explainable AI techniques for the data scientist to solve the data imbalance problems and train a better model without overfitting. The number in the majority class highly biases our classification accuracy. Although the parallel embeddings and the Morris sensitivity analysis demonstrate feature importance, features' interaction with the output, and other features, they fail to provide insights about model performance on imbalanced data. In the future, we plan to navigate other XAI tools and techniques for this analysis.

CHAPTER 7: Summary and Future Direction

7.1 Summary of the Thesis

This section presents the answers to the research questions and summarizes the contributions, limitations, and significant takeaways of the thesis.

7.1.1 Answers to the Research Questions based on the FIND Analysis

We discuss the three machine learning models in very detail from chapter 4 to chapter 6. These chapters demonstrate how each learning model satisfies a sequential process of student data engineering, learning model generation, and insight discovery in light of the FIND framework discussed in chapter 3. We will answer the research questions based on our analysis so far.

- ***RQ1.1: Leveraging the temporal nature of the student data, what do the different student data models capture?*** [Student Data Engineering]
  One of the significant gaps we try to fill in is that the learning analytics research has not yet adopted the usage of temporal analysis broadly. All of the three models are temporal. In our first model network analysis, we build weighted student networks that are cumulative by time. For example, if a student registers for two courses in the first semester with thirty other classmates, the edge weight for the course rule will be 30 for that student in the first semester. If the student registers for 40 more students in the second semester, the edge weight in the second semester will be 70. Here, we build our weighted network by utilizing and capturing the semester-wise time-series information. The fixed point feature is temporal too. We did predictive and statistical analysis on the network features extracted from the temporal weighted student network. Each

of the temporal network and fixpoint features was able to capture how the students behavior changes over time for both successful and at-risk students. For example, for the ego network density we proved that even with newer connections over time at-risk students networks do not get complete whereas successful students create complete ego networks very early in their career. The temporal nature of the student network data model helped to identify at-risk students earlier in their career whereas only using non-temporal academic features have a lower performance in identifying them.

The unsupervised clustering and the LSTM models for both classification and regression models use engineered temporal features too. We engineered some course-related semester-wise features to cluster the common patterns among student cohorts. The temporal projection of the course level progression captures how the at-risk and successful students attempted the courses and how it affects their performance.

For the LSTM RNN models for both classification and regression, SHAP explanations capture how each feature is relevant to a model outcome and how their variation or interaction over time changes the model output.

- ***RQ1.2: What is the benefit of addressing heterogeneity of high-volume student data?*** [Student Data Engineering]

  A data structure is called heterogeneous if it contains multiple sub-structures and various types and formats of data. Our data consist of temporal and non-temporal categorical and numeric variables. This heterogeneous student data from different data tables and sources are aggregated into weighted student networks, engineered for the unsupervised cluster and RNN analysis. Large volumes of aggregated heterogeneous student data can analyze and learn enormous amounts of unsupervised data. Meanwhile, raw data is primarily unlabeled and uncategorized. Moreover, heterogeneity aids in gaining a better understanding of vast amounts of data and allows for the effective use of such data in predictive modeling. Even though we maintain consistency while designing the feature vectors of each student for our analyses, it is typical for one student to be associated with numerous types of information, each of which corresponds to different perspectives. For example, only semester-wise course progression may not help us understand the students better. The additional nontemporal background or outcome(CGPA) information for a longitudinal volume of data helped us understand a typical pattern over time.

- ***RQ2: How do different explainable machine learning methods help produce better models of student success and risk?*** [Learning Models Generation]

  Detailed analyses of the explainable AI tools are provided in chapter 6. The SHAP explanations with force plots demonstrate how the features operate on the model for a time series analysis. Moreover, it further presents the pattern fluctuating for different cohorts. With the help of summary plots, we were able to see which features are essential, which have a constant effect on the model over time, and which features do not impact the model at all. The feature importance summary helps us select the relevant features for a proper model design. Therefore, SHAP is an ideal XAI tool for both domain experts and data scientists.

  Morris sensitivity analysis is an excellent tool for data scientists to understand and train the models better. It provides the feature importance and demonstrates how each feature correlates with the output variable by linearity or non-linearity.

  Visualization or visual reports are also a good form of explanation. The visualization of clusters for unsupervised K-means clustering provides a better understanding of the students' cohort, which is impossible to perceive with the tabular or non-visual representation.

- ***RQ3: How does each model support comprehending students' performance and effectively planning interventions? How may the insights gained from these models aid domain experts in obtaining actionable knowledge?*** [Insight Discovery]

  Each of the three models plays a different role in understanding the students better and provides more directions to further analyses. We believe each of them is important and serves various purposes to improve the research of Learning Analytics. For example, network analytics demonstrates how the collaboration among the students influences their performance. With the help of this analysis, domain experts will classify at-risk students who are prone to drop out or take a longer time to graduate earlier. Therefore, it can lead us to plan interventions to help at-risk students earlier in their academic careers. Faculty members can also incorporate the findings from this analysis to form teams in a course. For example, since we find the at-risk students are isolated and tend to linger, we can mix skill levels and expertise while forming teams so that no one will feel isolated and, therefore, perform worse than others. Moreover, the uniqueness of this model is that it does not depend on the social network or interaction logs data. It makes the data gathering process very convenient and consistent.

  The cluster analysis helps us find segregated patterns of different student cohorts and how they perform differently by projecting the temporal clusters from beginning to their eighth semesters.

  Even though the RNN models are black boxes, SHAP and Morris sensitivity analysis show us a path to interpret predictive results from any model. Moreover, they teach us how to iteratively train and diagnose a model's efficacy by validating feature interactions and importance. These explainable AI tools and techniques help the domain expert (SHAP analysis) trust the machine learning systems and incorporate their domain knowledge to verify these systems. They (Morris Sensitivity research) will also help the data scientists train and understand their implemented models better.

7.1.2 Contribution and Major Takeaways of the Thesis

The significant contribution of this dissertation is the three temporal analytical models implemented to gain deeper insights on students' performance and behavior analysis. The following three models are developed:

1. Weighted student network analytics [1]
2. Unsupervised k-means clustering [5]
3. Classification and regression analysis with LSTM recurrent neural network

Throughout the whole process, various temporal data models are investigated and implemented. We apply different preprocessing techniques, including feature selections and hyperparameter tuning on the same student data to prepare diverse student data models to address the input structure of the machine learning algorithms. Finally, we implement machine learning algorithms on these student data models to improve students' overall performance and learning experience. Each of the models teaches us various insights. For example, the temporal weighted student network analysis demonstrates how the students' connections influence the students' performance and behavior over time. The inclusion of network and fixed-point features improves early predicting the risky behavior of dropping out or longer time for graduation. The comprehensive insights from the model built from the offline database will help the domain experts to plan early interventions. Unsupervised cluster analysis allows us to see through the behavioral distinctions of successful and at-risk students throughout their academic character. The iterative process of building an RNN model teaches us lessons on practical model training and data. We try various hyperparameters and setups to develop a model with higher accuracy to predict the CGPAs ahead of time. The RNN models inform us how the lower data volume, imbalance, and irrelevant feature selection negatively influence performance accuracy. Explainable AI tools and techniques provide an insightful understanding of both the models and the student data.

We explore how to incorporate explainable AI and visual analytical results to assist the understanding mentioned above of the predictive models. Finally, we introduce discussions of different state-of-art explainable AI tools and techniques on our recurrent neural network models to demonstrate the efficacy of interpretability for actionable decision-making.

With the idea of incorporating more explainability in machine learning systems and tool design, I lead a prototype for an interactive dashboard to involve the adviser in the decision-making process. In this prototype, we propose incorporating a data model, an aggregate analytics view with visual and tabular explanations, and a view to understanding students individually with different attributes. I also develop a student data model in MongoDB, which addresses the heterogeneity of the student data with ease of usage. Moreover, the novelty of my research exhibits the importance of explainable AI for educators and policymakers to understand the model predictions. Finally, we envision optimizing learning analytics research outcomes and learning experiences by incorporating their expertise in the knowledge discovery and system design process.

We develop an empirical framework to structure the whole learning analytics process into actionable insights generation. We present a generalizable framework to demonstrate how the three different learning models help the domain experts and educational policymakers identify students' learning patterns and design actionable insights.

7.1.3 Limitations

As CS educational researchers and practitioners, we struggle the most in gaining access to the varying data sources throughout the research. For example, our research on team collaboration with network analytics would be more effective if we could gather more data on students' demography, card swipe, dormitory, housing, library, food, and counseling services. The sources for these data are sometimes unknown, and very hard to initialize the data gathering process for strict protocols. We understand the privacy of the student data is a significant concern. However, it is not well documented where and how to seek help and guidance, especially if it has a larger staff population. Most of the time, it takes a very lengthy amount of time by which a researcher or Ph.D. student may change the job or graduate. Moreover, as a team, we struggle to understand the existing features of the data due to the absence of proper institutional documentation.

Inconsistent use of LMS and instructional methods leads to data sparsity. Moreover, information on the admission, background, and demography of the students is also very sparse. We believe this information will help us to gather more actionable insights. This information will help us guide the diversity of students who struggle mainly for being a minority or being exceptional demographically.

Learning analytical research is a longitudinal process. From our deep analysis of the RNN models, we see how a higher volume of data is necessary to build an effective predictive model. However, due to a frequent change in the curriculum, we cannot gather data for a longer time since it will include inconsistencies in the data.

Despite the limitations, I believe CS education research and AI in education is one of the most vibrant research areas where we can be involved in knowledge discovery and implementation and improve human's (students') lives directly.

7.2 Future Directions

Throughout the research, experience, and extensive survey, it is established that the learning analytics research focuses on three primary areas (see figure 7.1) â

1. Journey Analytics

2. Self-regulated learning, and

3. Leveraging machine learning and AI tools and techniques into the LMS systems and learning experience

We leverage the experience and knowledge on machine learning and AI tools and techniques to do journey analytics with the semester progressions. For example, we develop the EAGER ANALYTICS tool for Degree Analytics, which analyzes the degree progression journey throughout the semesters. As part of this, self-regulated learning can be one of the future researches. It's a cyclical procedure in which a student sets a task, checks their progress, and then evaluates them. The cycle then repeats itself as the learner adjusts and prepares for the next challenge using reflection. The concept of the prototype we develop already involves self-regulated learning for the policymakers and domain experts. We regulate their experience through multiple user studies. The iterative development of the tools and their composition is based on the feedback from the user studies.

The research group did some preliminary research on the sentiment analysis on weekly reflections of intro-level programming courses. Researchers find a high correlation between the weekly reviews and students' performance in the upcoming weeks. The sentiment analysis on students' observations of course and an instructor can help design early intervention plans for course analytics. A plan to research and develop an automated early alert system for that specific course to minimize the human interruption in the intervention process can use the reflections from the student, which

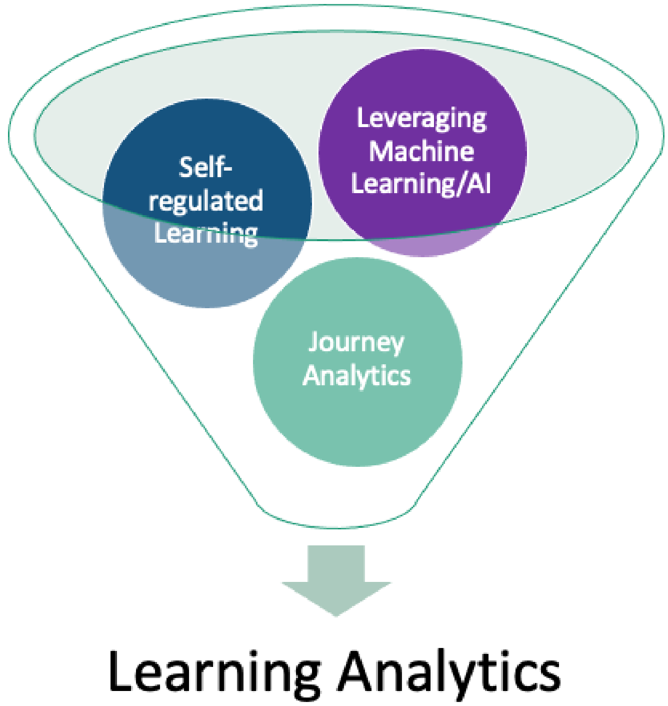


Figure 7.1: Three Major Areas of Learning Analytics Research

will not be biased. It will positively drive the students to think about a course, meet and interact with the TAs and instructors more frequently and effectively. This will be an excellent area for further research work. Both the graduate and undergraduate students can participate in the knowledge discovery process by actively participating in the research and gathering more valuable and exciting data and tool development to further the research.

The insights and the knowledge from the network analysis can be leveraged to design effective team collaboration. To continue this research, we need longitudinal analysis, user studies, and experimentations. After the publication of the network analytics model, surprisingly, we found that **Texas AM University** is already leading a **peer teaching (PT)** program based on the same concept derived from our mathematical analysis [137]. Peer teachers **(PTs)** are high-achieving undergraduates who want to assist others in achieving in the classes in which they excelled earlier.

The PTs assist students in comprehending material given in classrooms and labs, practicing computational thinking and programming skills, and selecting appropriate courses. Many students prefer to seek assistance from PTs rather than professors and teaching assistants because they feel better connected to them both academically and socially.

Since analytics and data mining experiments in education are becoming increasingly well-known, there is more competition to gain research prospects and implement practical applications. I plan to explore the research area with realistic plans and purposes.

REFERENCES

[1] N. Nur, N. Park, M. Dorodchi, W. Dou, M. J. Mahzoon, X. Niu, and M. L. Maher, "Student network analysis: a novel way to predict delayed graduation in higher education," in *International Conference on Artificial Intelligence in Education*, pp. 370–382, Springer, 2019.

[2] J. Han, M. Kamber, and J. Pei, "Data mining concepts and techniques, third edition," 2012.

[3] M. J. Mahzoon, M. Maher, O. Eltayeby, W. Dou, and K. Grace, "A sequence data model for analyzing temporal patterns of student data," *Journal of Learning Analytics*, vol. 5, pp. 55–74, 04 2018.

[4] M. A. Javed, M. S. Younis, S. Latif, J. Qadir, and A. Baig, "Community detection in networks: A multidisciplinary review," *Journal of Network and Computer Applications*, vol. 108, pp. 87–111, 2018.

[5] A. Al-Doulat, N. Nur, A. Karduni, A. Benedict, E. Al-Hossami, M. L. Maher, W. Dou, M. Dorodchi, and X. Niu, "Making sense of student success and risk through unsupervised machine learning and interactive storytelling," in *International Conference on Artificial Intelligence in Education*, pp. 3–15, Springer, 2020.

[6] J. Zhang, W. Chen, M. Gao, and G. Shen, "K-means-clustering-based fiber nonlinearity equalization techniques for 64-qam coherent optical communication system," *Optics express*, vol. 25, no. 22, pp. 27570–27580, 2017.

[7] A. S. López and F. Martnez, "Determination of the number of cluster a priori using a k-means algorithm," *Int. J. Eng. Tech. Res*, vol. 6, pp. 93–95, 2016.

[8] T. K. Vintsyuk, "Speech discrimination by dynamic programming," *Cybernetics*, vol. 4, no. 1, pp. 52–57, 1968.

[9] S. An, "Recurrent Neural Networks." https://www.cc.gatech.edu/ san37/post/dlhc-rnn/. [Online; accessed 04/22/2019].

[10] "TensorFlow: Time Series Forecasting." https://www.tensorflow.org/tutorials/. [Online; accessed 04/22/2019].

[11] K. Kawakami, "Supervised sequence labelling with recurrent neural networks," *Ph. D. thesis*, 2008.

[12] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

[13] D. Gašević, V. Kovanović, and S. Joksimović, "Piecing the learning analytics puzzle: A consolidated model of a field of research and practice," *Learning: Research and Practice*, vol. 3, no. 1, pp. 63–78, 2017.

[14] A. Coates, A. Karpathy, and A. Ng, "Emergence of object-selective features in unsupervised feature learning," *Advances in Neural Information Processing Systems*, vol. 25, pp. 2681–2689, 2012.

[15] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of big data*, vol. 2, no. 1, pp. 1–21, 2015.

[16] K. E. Arnold and M. D. Pistilli, "Course signals at purdue: Using learning analytics to increase student success," in *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, LAK '12, (New York, NY, USA), p. 267â270, Association for Computing Machinery, 2012.

[17] M. Kapur, "Temporality matters: Advancing a method for analyzing problem-solving processes in a computer-supported collaborative environment," *International Journal of Computer-Supported Collaborative Learning*, vol. 6, pp. 39–56, 2011.

[18] P. Reimann, "Time is precious: Variable- and event-centred approaches to process analysis in cscl research," *I. J. Computer-Supported Collaborative Learning*, vol. 4, pp. 239–257, 09 2009.

[19] C. Conati, K. Porayska-Pomsta, and M. Mavrikis, "Ai in education needs interpretable machine learning: Lessons from open learner modelling," *arXiv preprint arXiv:1807.00154*, 2018.

[20] D. Gasevic, V. Kovanovic, and S. Joksimovic, "Piecing the learning analytics puzzle: a consolidated model of a field of research and practice," *Learning: Research and Practice*, vol. 3, no. 1, pp. 63–78, 2017.

[21] T. Elias, "Learning analytics," *Learning*, pp. 1–22, 2011.

[22] N. Sclater, *Learning analytics explained.* Routledge, 2017.

[23] K. Sin and L. Muthu, "Application of big data in education data mining and learning analytics-a literature review," in *SOCO 2015*, 2015.

[24] M. A. Chatti, A. L. Dyckhoff, U. Schroeder, and H. Thüs, "A reference model for learning analytics," *Int. J. Technol. Enhanc. Learn.*, vol. 4, p. 318â331, Jan. 2012.

[25] A. G. Picciano *et al.*, "Beyond student perceptions: Issues of interaction, presence, and performance in an online course," *Journal of Asynchronous learning networks*, vol. 6, no. 1, pp. 21–40, 2002.

[26] H. Drachsler and W. Greller, “The pulse of learning analytics understandings and expectations from the stakeholders,” in *Proceedings of the 2nd international conference on learning analytics and knowledge*, pp. 120–129, 2012.

[27] S. Ray, “Big data in education,” *2321-0451*, pp. 6–8, 12 2013.

[28] P. Blikstein, “Using learning analytics to assess students’ behavior in open-ended programming tasks,” in *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, LAK ’11, (New York, NY, USA), p. 110â116, Association for Computing Machinery, 2011.

[29] C. Harvey, “Top open source tools for big data,” *Datamation*, 2012.

[30] M. E. Zorrilla, E. Menasalvas, D. Marín, E. Mora, and J. Segovia, “Web usage mining project for improving web-based learning sites,” in *Computer Aided Systems Theory – EUROCAST 2005* (R. Moreno Díaz, F. Pichler, and A. Quesada Arencibia, eds.), (Berlin, Heidelberg), pp. 205–210, Springer Berlin Heidelberg, 2005.

[31] Ãngel HernÃ¡ndez-GarcÃa, I. GonzÃ¡lez-GonzÃ¡lez, A. I. JimÃ©nez-Zarco, and J. Chaparro-PelÃ¡ez, “Applying social learning analytics to message boards in online distance learning: A case study,” *Computers in Human Behavior*, vol. 47, pp. 68–80, 2015. Learning Analytics, Educational Data Mining and data-driven Educational Decision Making.

[32] Ãngel HernÃ¡ndez-GarcÃa, I. GonzÃ¡lez-GonzÃ¡lez, A. JimÃ©nez-Zarco, and J. Chaparro-PelÃ¡ez, “Visualizations of online course interactions for social network learning analytics,” *International Journal of Emerging Technologies in Learning (iJET)*, vol. 11, no. 07, pp. 6–15, 2016.

[33] B. Schreurs, C. Teplovs, R. Ferguson, M. de Laat, and S. Buckingham Shum, “Visualizing social learning ties by type and topic: Rationale and concept demonstrator,” in *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, LAK ’13, (New York, NY, USA), p. 33â37, Association for Computing Machinery, 2013.

[34] A. F. Wise, “Designing pedagogical interventions to support student use of learning analytics,” in *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge*, LAK ’14, (New York, NY, USA), p. 203â211, Association for Computing Machinery, 2014.

[35] M. J. RodrÃguez-Triana, A. MartÃnez-MonÃ©s, J. I. Asensio-PÃ©rez, and Y. Dimitriadis, “Scripting and monitoring meet each other: Aligning learning analytics and learning design to support teachers in orchestrating cscl situations,” *British Journal of Educational Technology*, vol. 46, no. 2, pp. 330–343, 2015.

[36] S. de Freitas, D. Gibson, C. Du Plessis, P. Halloran, E. Williams, M. Ambrose, I. Dunwell, and S. Arnab, "Foundations of dynamic learning analytics: Using university student data to increase retention," *British Journal of Educational Technology*, vol. 46, no. 6, pp. 1175–1188, 2015.

[37] O. E. Tsivitanidou and C. P. Constantinou, "A study of students' heuristics and strategy patterns in web-based reciprocal peer assessment for science learning," *The Internet and Higher Education*, vol. 29, pp. 12–22, 2016.

[38] K. Kitto, S. Cross, Z. Waters, and M. Lupton, "Learning analytics beyond the lms: the connected learning analytics toolkit," *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, 2015.

[39] M. Akcaoglu and N. D. Bowman, "Using instructor-led facebook groups to enhance studentsâ perceptions of course content," *Computers in Human Behavior*, vol. 65, pp. 582–590, 2016.

[40] C. Romero, S. Ventura, and E. García, "Data mining in course management systems: Moodle case study and tutorial," *Computers & Education*, vol. 51, no. 1, pp. 368–384, 2008.

[41] S. M. Jayaprakash, E. W. Moody, E. J. Lauría, J. R. Regan, and J. D. Baron, "Early alert of academically at-risk students: An open source analytics initiative," *Journal of Learning Analytics*, vol. 1, no. 1, pp. 6–47, 2014.

[42] A. Essa and H. Ayad, "Student success system: Risk analytics and data visualization using ensembles of predictive models," in *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, LAK '12, (New York, NY, USA), p. 158â161, Association for Computing Machinery, 2012.

[43] K. T. Chui, D. C. L. Fung, M. D. Lytras, and T. M. Lam, "Predicting at-risk university students in a virtual learning environment via a machine learning algorithm," *Computers in Human Behavior*, vol. 107, p. 105584, 2020.

[44] S. P. Choi, S. S. Lam, K. C. Li, and B. T. 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.

[45] A. Wolff, Z. Zdrahal, D. Herrmannova, J. Kuzilek, and M. Hlosta, "Developing predictive models for early detection of at-risk students on distance learning modules," 2014.

[46] C. Monroy, V. S. Rangel, and R. Whitaker, "A strategy for incorporating learning analytics into the design and evaluation of a k-12 science curriculum.," *Journal of Learning Analytics*, vol. 1, no. 2, pp. 94–125, 2014.

[47] C.-Y. Chou, S.-F. Tseng, W.-C. Chih, Z.-H. Chen, P.-Y. Chao, K. R. Lai, C.-L. Chan, L.-C. Yu, and Y.-L. Lin, "Open student models of core competencies at the curriculum level: Using learning analytics for student reflection," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 1, pp. 32–44, 2015.

[48] V. Snodgrass Rangel, E. R. Bell, C. Monroy, and J. R. Whitaker, "Toward a new approach to the evaluation of a digital curriculum using learning analytics," *Journal of Research on Technology in Education*, vol. 47, no. 2, pp. 89–104, 2015.

[49] M. Dorodchi, A. Benedict, D. Desai, M. J. Mahzoon, S. MacNeil, and N. Dehbozorgi, "Design and implementation of an activity-based introductory computer science course (cs1) with periodic reflections validated by learning analytics," in *2018 IEEE Frontiers in Education Conference (FIE)*, pp. 1–8, IEEE, 2018.

[50] A. L. Dyckhoff, D. Zielke, M. Bültmann, M. A. Chatti, and U. Schroeder, "Design and implementation of a learning analytics toolkit for teachers.," *J. Educ. Technol. Soc.*, vol. 15, no. 3, pp. 58–76, 2012.

[51] B. Dietz-Uhler and J. E. Hurn, "Using learning analytics to predict (and improve) student success: A faculty perspective," *Journal of interactive online learning*, vol. 12, no. 1, pp. 17–26, 2013.

[52] E. Demeter, M. Dorodchi, E. Al-Hossami, A. Benedict, L. Walker, and J. Smail, "Predicting time-to-degree of first time in college students." Submitted, under review., 2021.

[53] B. Amnueypornsakul, S. Bhat, and P. Chinprutthiwong, "Predicting attrition along the way: The uiuc model," in *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pp. 55–59, 2014.

[54] D. Ifenthaler and C. Widanapathirana, "Development and validation of a learning analytics framework: Two case studies using support vector machines," *Technology, Knowledge and Learning*, vol. 19, no. 1-2, pp. 221–240, 2014.

[55] M. d'Aquin and N. Jay, "Interpreting data mining results with linked data for learning analytics: motivation, case study and directions," in *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pp. 155–164, 2013.

[56] J. M. Kleinberg *et al.*, "Authoritative sources in a hyperlinked environment.," in *SODA*, vol. 98, pp. 668–677, Citeseer, 1998.

[57] M. Saqr and J. Nouri, "High resolution temporal network analysis to understand and improve collaborative learning," in *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, pp. 314–319, 2020.

[58] A. Ahadi, V. Behbood, A. Vihavainen, J. Prior, and R. Lister, "Students' syntactic mistakes in writing seven different types of sql queries and its application

to predicting students' success," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pp. 401–406, 2016.

[59] N. Dehbozorgi and S. MacNeil, "Semi-automated analysis of reflections as a continuous course," in *2019 IEEE Frontiers in Education Conference (FIE)*, pp. 1–5, IEEE, 2019.

[60] A. Ezen-Can, K. E. Boyer, S. Kellogg, and S. Booth, "Unsupervised modeling for understanding mooc discussion forums: a learning analytics approach," in *Proceedings of the fifth international conference on learning analytics and knowledge*, pp. 146–150, 2015.

[61] J. M. Reilly and B. Schneider, "Predicting the quality of collaborative problem solving through linguistic analysis of discourse.," *International Educational Data Mining Society*, 2019.

[62] S. Sorour, K. Goda, and T. Mine, "Correlation of topic model and student grades using comment data mining," in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pp. 441–446, 2015.

[63] T. D. Ullmann, "Automated analysis of reflection in writing: Validating machine learning approaches," *International Journal of Artificial Intelligence in Education*, vol. 29, no. 2, pp. 217–257, 2019.

[64] Z. Ren, X. Ning, A. S. Lan, and H. Rangwala, "Grade prediction based on cumulative knowledge and co-taken courses.," *International Educational Data Mining Society*, 2019.

[65]

[66] H. Jiang, M. Iandoli, S. Van Dessel, S. Liu, and J. Whitehill, "Measuring students' thermal comfort and its impact on learning.," *International Educational Data Mining Society*, 2019.

[67] I. Hilliger, C. Miranda, S. Celis, and M. Pérez-SanAgustín, "Evaluating usage of an analytics tool to support continuous curriculum improvement.," in *EC-TEL (Practitioner Proceedings)*, 2019.

[68] S. M. MacNeil, M. M. Dorodchi, E. Al-Hossami, A. Benedict, D. Desai, and M. J. Mahzoon, "Curri: A curriculum visualization system that unifies curricular dependencies with temporal student data," in *2020 ASEE Virtual Annual Conference Content Access*, 2020.

[69] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper, "Recommender systems in technology enhanced learning," in *Recommender systems handbook*, pp. 387–415, Springer, 2011.

[70] M. Mohseni, M. L. Maher, K. Grace, N. Najjar, F. Abbas, and O. Eltayeby, "Pique: Recommending a personalized sequence of research papers to engage student curiosity," in *International Conference on Artificial Intelligence in Education*, pp. 201–205, Springer, 2019.

[71] J. P. Campbell, *Utilizing student data within the course management system to determine undergraduate student academic success: An exploratory study*. Purdue University, 2007.

[72] J. Heinrich and D. Weiskopf, "State of the art of parallel coordinates.," in *Eurographics (State of the Art Reports)*, pp. 95–116, 2013.

[73] S. Lipovetsky and M. Conklin, "Analysis of regression in game theory approach," *Applied Stochastic Models in Business and Industry*, vol. 17, no. 4, pp. 319–330, 2001.

[74] W. W. S. Wei, "Time series analysis - univariate and multivariate methods," 1989.

[75] W. Vandaele, *Applied time series and Box-Jenkins models / Walter Vandaele*. New York: Academic Press, 1983 - 1983.

[76] I. Molenaar, "Advances in temporal analysis in learning and instruction.," *Frontline Learning Research*, vol. 2, no. 4, pp. 15–24, 2014.

[77] M. Fei and D.-Y. Yeung, "Temporal models for predicting student dropout in massive open online courses," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pp. 256–263, IEEE, 2015.

[78] S. Tang, J. C. Peterson, and Z. A. Pardos, "Deep neural networks and how they apply to sequential education data," in *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pp. 321–324, 2016.

[79] B. Jeon, E. Shafran, L. Breitfeller, J. Levin, and C. P. Rosé, "Time-series insights into the process of passing or failing online university courses using neural-induced interpretable student states," *arXiv preprint arXiv:1905.00422*, 2019.

[80] S. M. Lundberg, B. Nair, M. S. Vavilala, M. Horibe, M. J. Eisses, T. Adams, D. E. Liston, D. K.-W. Low, S.-F. Newman, J. Kim, *et al.*, "Explainable machine-learning predictions for the prevention of hypoxaemia during surgery," *Nature biomedical engineering*, vol. 2, no. 10, pp. 749–760, 2018.

[81] B. Mittelstadt, C. Russell, and S. Wachter, "Explaining explanations in ai," in *Proceedings of the conference on fairness, accountability, and transparency*, pp. 279–288, 2019.

[82] D. Wang, Q. Yang, A. Abdul, and B. Y. Lim, "Designing theory-driven user-centric explainable ai," in *Proceedings of the 2019 CHI conference on human factors in computing systems*, pp. 1–15, 2019.

[83] B. Nushi, E. Kamar, and E. Horvitz, "Towards accountable ai: Hybrid human-machine analyses for characterizing system failure," in *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, vol. 6, 2018.

[84] P. Madumal, T. Miller, F. Vetere, and L. Sonenberg, "Towards a grounded dialog model for explainable artificial intelligence," *arXiv preprint arXiv:1806.08055*, 2018.

[85] A. Kirsch, "Explain to whom? putting the user in the center of explainable ai," in *Proceedings of the First International Workshop on Comprehensibility and Explanation in AI and ML 2017 co-located with 16th International Conference of the Italian Association for Artificial Intelligence (AI* IA 2017)*, 2017.

[86] B. Goodman and S. Flaxman, "European union regulations on algorithmic decision-making and a âright to explanationâ," *AI magazine*, vol. 38, no. 3, pp. 50–57, 2017.

[87] E. Musk and W. Isaacson, "Artificial Intelligence Could Wipe Out Humanity." https://sagaciousnewsnetwork.com/elon-musk-artificial-intelligencecould-wipe-out-humanty/, 2014. [Online; accessed 04/22/2019].

[88] E. Musk, "AI âvastly more risky than North Koreaâ ." https://twitter.com/elonmusk/status/896166762361704450, 2017. [Online; accessed 04/22/2019].

[89] B. Waltl and R. Vogl, "Explainable artificial intelligence: The new frontier in legal informatics," *Jusletter IT*, vol. 4, pp. 1–10, 2018.

[90] R. Cellan-Jones, "Stephen hawking warns artificial intelligence could end mankind," *BBC news*, vol. 2, no. 10, p. 2014, 2014.

[91] L. Corrin and P. De Barba, "Exploring students' interpretation of feedback delivered through learning analytics dashboards," in *31st Annual Conference of the Australian Society for Computers in Tertiary Education (ASCILITE 2014),'Rhetoric and Reality: Critical perspectives on educational technology', Dunedin, New Zealand, 23-26 November 2014/B. Hegarty, J. McDonald and SK Loke (eds.)*, pp. 629–633, 2014.

[92] S. Bull and H. Pain, *" Did I Say What I Think I Said, and Do You Agree with Me?": Inspecting and Questioning the Student Model.* University of Edinburgh, Department of Artificial Intelligence, 1995.

[93] S. Bull and J. Kay, "Smiliâº: A framework for interfaces to learning data in open learner models, learning analytics and related fields," *International Journal of Artificial Intelligence in Education*, vol. 26, no. 1, pp. 293–331, 2016.

[94] M. De Laat and F. R. Prinsen, "Social learning analytics: Navigating the changing settings of higher education.," *Research & Practice in Assessment*, vol. 9, pp. 51–60, 2014.

[95] W. Greller and H. Drachsler, "Translating learning into numbers: A generic framework for learning analytics," *Journal of Educational Technology & Society*, vol. 15, no. 3, pp. 42–57, 2012.

[96] "Banner UNCC." https://servicecatalog.uncc.edu/service/reporting-and-analytics-systems . [Online; accessed 04/22/2019].

[97] K. E. Arnold, "Signals: Applying academic analytics.," *Educause Quarterly*, vol. 33, no. 1, p. n1, 2010.

[98] S. M. Jayaprakash, E. W. Moody, E. J. Lauría, J. R. Regan, and J. D. Baron, "Early alert of academically at-risk students: An open source analytics initiative," *Journal of Learning Analytics*, vol. 1, no. 1, pp. 6–47, 2014.

[99] C. Romero, M.-I. López, J.-M. Luna, and S. Ventura, "Predicting students' final performance from participation in on-line discussion forums," *Computers & Education*, vol. 68, pp. 458–472, 2013.

[100] J. P. Campbell, P. B. DeBlois, and D. G. Oblinger, "Academic analytics: A new tool for a new era," *EDUCAUSE review*, vol. 42, no. 4, p. 40, 2007.

[101] P. Jaccard, "The distribution of the flora in the alpine zone. 1," *New phytologist*, vol. 11, no. 2, pp. 37–50, 1912.

[102] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social networks*, vol. 1, no. 3, pp. 215–239, 1978.

[103] A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani, "The architecture of complex weighted networks," *Proceedings of the national academy of sciences*, vol. 101, no. 11, pp. 3747–3752, 2004.

[104] S. Wasserman, K. Faust, *et al.*, "Social network analysis: Methods and applications," 1994.

[105] J. M. Anthonisse, "The rush in a directed graph," *Stichting Mathematisch Centrum. Mathematische Besliskunde*, no. BN 9/71, 1971.

[106] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[107] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web.," tech. rep., Stanford InfoLab, 1999.

[108] Y. Li, O. Martinez, X. Chen, Y. Li, and J. E. Hopcroft, "In a world that counts: Clustering and detecting fake social engagement at scale," in *Proceedings of the 25th International Conference on World Wide Web*, pp. 111–120, 2016.

[109] C. Mavroforakis, I. Valera, and M. Gomez-Rodriguez, "Modeling the dynamics of learning activity on the web," in *Proceedings of the 26th International Conference on World Wide Web*, pp. 1421–1430, 2017.

[110] M. X. Hoang, X.-H. Dang, X. Wu, Z. Yan, and A. K. Singh, "Gpop: Scalable group-level popularity prediction for online content in social networks," in *Proceedings of the 26th international conference on world wide web*, pp. 725–733, 2017.

[111] S. Chaturvedi, V. Castelli, R. Florian, R. M. Nallapati, and H. Raghavan, "Joint question clustering and relevance prediction for open domain non-factoid question answering," in *Proceedings of the 23rd international conference on World wide web*, pp. 503–514, 2014.

[112] J. Xie, B. K. Szymanski, and X. Liu, "Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process," in *2011 ieee 11th international conference on data mining workshops*, pp. 344–349, IEEE, 2011.

[113] R. S. Burt, "Models of network structure," *Annual review of sociology*, vol. 6, no. 1, pp. 79–141, 1980.

[114] R. Abraham and J. E. Marsden, *Foundations of mechanics*. No. 364, American Mathematical Soc., 2008.

[115] D. Gibson, J. Kleinberg, and P. Raghavan, "Inferring web communities from link topology," in *Proceedings of the ninth ACM conference on Hypertext and hypermedia: links, objects, time and space—structure in hypermedia systems: links, objects, time and space—structure in hypermedia systems*, pp. 225–234, 1998.

[116] J. C. Miller, G. Rae, F. Schaefer, L. A. Ward, T. LoFaro, and A. Farahat, "Modifications of kleinberg's hits algorithm using matrix exponentiation and web log records," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 444–445, 2001.

[117] L. Li, Y. Shang, and W. Zhang, "Improvement of hits-based algorithms on web documents," in *Proceedings of the 11th international conference on World Wide Web*, pp. 527–535, 2002.

[118] C. Kang, N. Park, B. A. Prakash, E. Serra, and V. S. Subrahmanian, "Ensemble models for data-driven prediction of malware infections," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pp. 583–592, 2016.

[119] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, and V. Subrahmanian, "Rev2: Fraudulent user prediction in rating platforms," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 333–341, 2018.

[120] D. P. Diaz and R. B. Cartnal, "Students' learning styles in two classes: Online distance learning and equivalent on-campus," *College teaching*, vol. 47, no. 4, pp. 130–135, 1999.

[121] N. C. for Education Statistics, "Table 326.10.," 2016.

[122] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 559–563, 2017.

[123] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social networks*, vol. 1, no. 3, pp. 215–239, 1978.

[124] A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani, "The architecture of complex weighted networks," *Proceedings of the national academy of sciences*, vol. 101, no. 11, pp. 3747–3752, 2004.

[125] K. E. Joyce, P. J. Laurienti, J. H. Burdette, and S. Hayasaka, "A new measure of centrality for brain networks," *PloS one*, vol. 5, no. 8, p. e12200, 2010.

[126] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, Oakland, CA, USA, 1967.

[127] A. Ng, "Clustering with the k-means algorithm," *Machine Learning*, 2012.

[128] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series.," in *KDD workshop*, vol. 10, pp. 359–370, Seattle, WA, USA:, 1994.

[129] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the royal statistical society: series B (statistical methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[130] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[131] C. Lai, M. J. Reinders, and L. Wessels, "Random subspace method for multivariate feature selection," *Pattern recognition letters*, vol. 27, no. 10, pp. 1067–1076, 2006.

[132] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st international conference on neural information processing systems*, pp. 4768–4777, 2017.

[133] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial intelligence*, vol. 267, pp. 1–38, 2019.

[134] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowledge and information systems*, vol. 41, no. 3, pp. 647–665, 2014.

[135] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International Conference on Machine Learning*, pp. 3145–3153, PMLR, 2017.

[136] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola, "Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index," *Computer physics communications*, vol. 181, no. 2, pp. 259–270, 2010.

[137] "PEER TEACHERS." https://engineering.tamu.edu/cse/academics/peer-teachers/index.html/. [Online; accessed 04/22/2019].

APPENDIX A: List of Features & Eager Learning Analytics

A.1 Features Used in RNN

Table A.1: Numeric vs Categorical features

Numeric Features	**Categorical Features**
GPA Credits	
Credits Attempted	
Credits Passed	
Minimum Required Credits	
Maximum Required Credits	
Engineered Features-	GPA Type
1. Percentage of F Credits	Student Population Description
2. Percentage of W Credits	Current Time Status Description
3. Percentage of Transfered Credits	Employment Type Description
4. Percentage of A Credits	
5. Percentage of B Credits	
6. Percentage of C Credits	
7. Percentage of D Credits	
8. Course Level Progression	

Some of the features are common among the features mentioned below since we try to deduct the unimportant and irrelevant features for the analytical model in different ways.

Table A.2: Features Removed for Sparsity and by Feature Engineering from our Analysis

Sparse Features	**Features removed by Engineering**
Spouse Accompanied Indication	College
Children Number	Class size
Native Language	Enrolled Indication
Nation of Citizenship	Enrollment Status
Marital Status	Academic Standing
Primary Ethnicity	Student Status
Grad Academic Period Intended	Department
Withdrawal Reason	Current Time Status
Class Rank	Student Population
School Rank	Major
College	Admissions Population
Previous School Percentile	Progress Evaluation
School	Residency Indication
School GPA	Gender
Hours Attended,	Quality Points
Course Title Long	GPA
Veteran Category	

A.2 Eager Learning Analytics : Interactive Learning Analytics Tool

We used paper prototypes first to brainstorm on the tool and then used "Indigo Design Studio" to build our prototype tool. I led the design of this prototype. Another Ph.D. student from our learning analytics research group, Aileen Benedict, collaborated with me to design the "Indigo Design Studio" to do our pilot study on the prototype. We used paper prototypes first to brainstorm on the tool and then used

"Indigo Design Studio" to build our prototype tool.

I collaborated with two other Ph.D. students from our group to develop the preliminary version of the tool. A past Ph.D. student, Dr. Alireza Karduni, built the tool from scratch with python, React Javascript framework, and MongoDB database. I collaborated with Dr. Karduni on implementing and generating results for the second module - aggregated analytics. I also implemented an optimal data model formatted as a JSON document in the MongoDB database.

Since the tool is a team effort and ultimately implemented by others (the majority of its features), I did not discuss the tool in the main content of the dissertation. My main contribution to this tool are :

1. Leading the design of the prototype
2. Designing and Implementing the backend of the tool in MongoDB
3. Implementing the Aggregate Analytics and incorporating the results into the tool

The current design of the "Eager Analytics" tool is different from the design we will discuss here. Another Ph.D. student-led the "Individual Story" module and made it the tool's primary focus. The current tool is now completely ready to use professionally.

The following version of the tool is the first published version I work on [5]. This version is also associated with my second analytical model discussed in section 5.2.1. Our team also conducted a user study with this analytical model for the publication [5]. "Eager Analytics" mainly consists of three main views.

1. Student Data Model
2. Aggregated Analytics
3. Individual Story

A.2.1 Student Data Model

There are three sets of available features in the student data model from which we will select features for each respective node (See Figure A.1). The nodes are 1) the background features which consist of demographic, admission and background information of the students, 2) the semester features, which consist of semester-wise information such as all the course information, cumulative GPA, class rank, major, progress evaluation, etc. 3) Outcome features which consist of the features for performance evaluation such as GPA, number of years of enrollment, etc. Users have to click on each corresponding feature block, and the features will be added to the respective selected block. Users can choose a feature as a salient or contextual feature. By default, all the features are contextual. Users can choose features as salient by clicking on the check box on the right side of each feature in the Selected Features blocks.

There is a navigation panel at the bottom of each view of the prototype to provide the option to navigate back and forth.

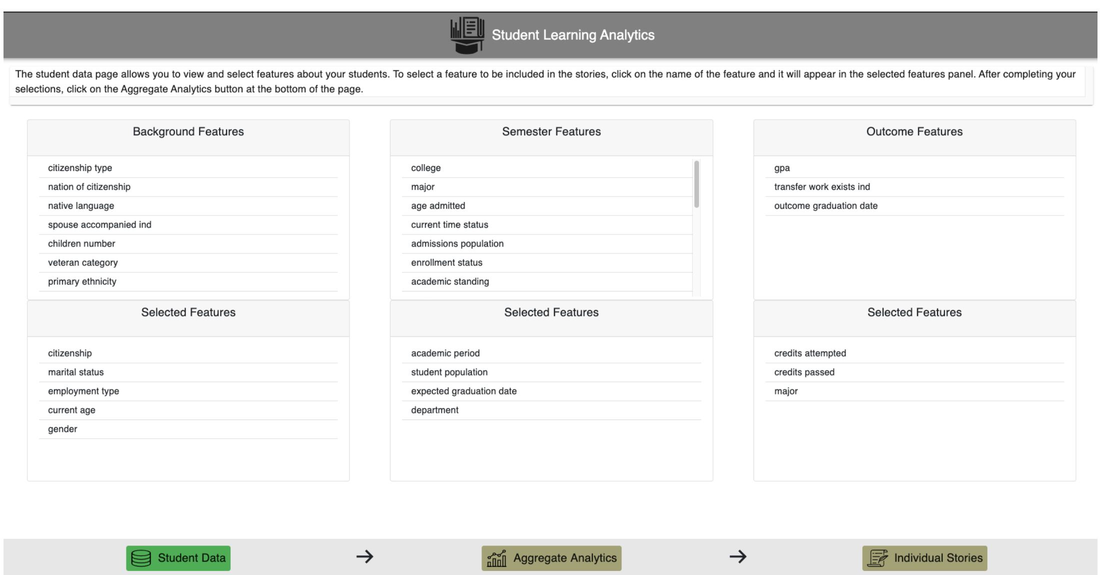


Figure A.1: Student Data Model (After Selection)

A.2.2 Aggregated Analytics

The aggregated analytics view is mainly divided into two parts. The first part is the left panel of the page, where there is a configuration module. Users need to configure some filters according to their needs from the configurations module. For example, it has three filters representing different features users want to navigate, cluster statistics for certain features (currently just showing GPA) and a GPA threshold for analytics. We are doing the analytics for only CGPA threshold 2. For example, if a user selects GPA as a success measure, they can regulate the threshold scale according to the need of their query. If they choose the threshold as 2, the aggregated analytics will be shown, considering successful students will have a GPA of greater or equal to 2 and vice versa.

In the "Aggregate Analysis", at the bottom of the page, we are showing a table. This table represents the values of the selected features (which are picked earlier in the data model) to evaluate the decision of being successful or at-risk. Each row of this table is presenting information on a single student. And the rows are colored based on the clusters of the students derived by the aggregate analytics in the cluster analysis. In the top part of this page, there are two graphs. The leftmost graph shows the results based on the K-means clusters we discussed in section 5.2.1. The clusters are colored differently. While selecting a student from the table, the corresponding cluster for that selected student is highlighted in this graph. The rightmost cluster shows a box plot distribution of the CGPA in the clusters.

We can select some students for further analysis in the individual story tab. The tool will generate the stories of the selected students on the "Individual Story" page.

A.2.3 Individual Story

On the left side of this view, there is a list of the selected students from the aggregated analytics page. The top right corner of the page shows three types of

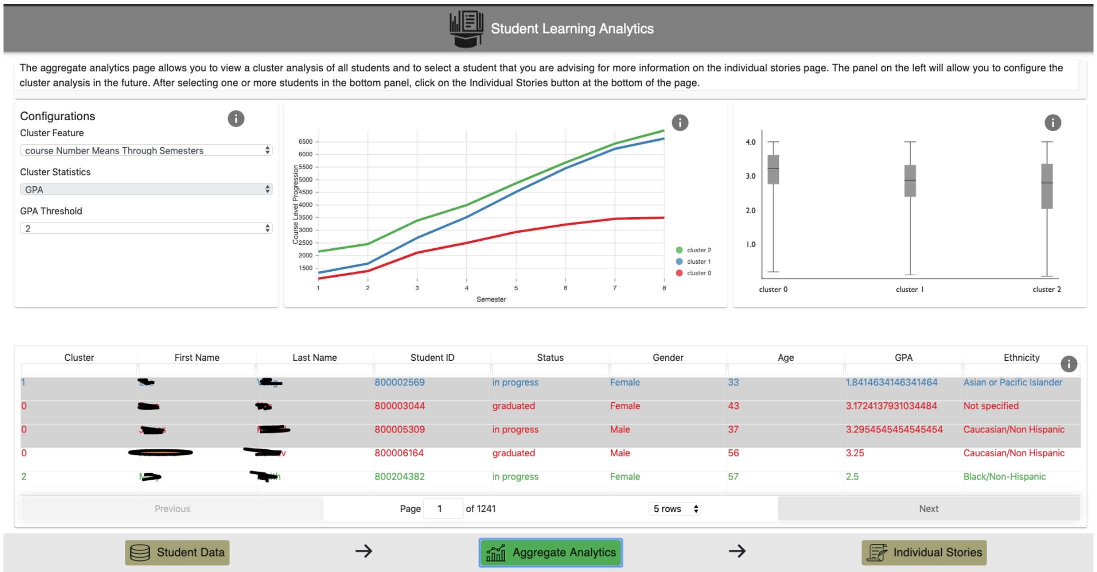


Figure A.2: Aggregate Analysis

stories for the students chosen. 1) "Default" stories are based on pre-fixed features, 2) "Temporal" stories are generated by the selected features in the student data model, and 3) "Outcome" stories are generated for the outcome features.

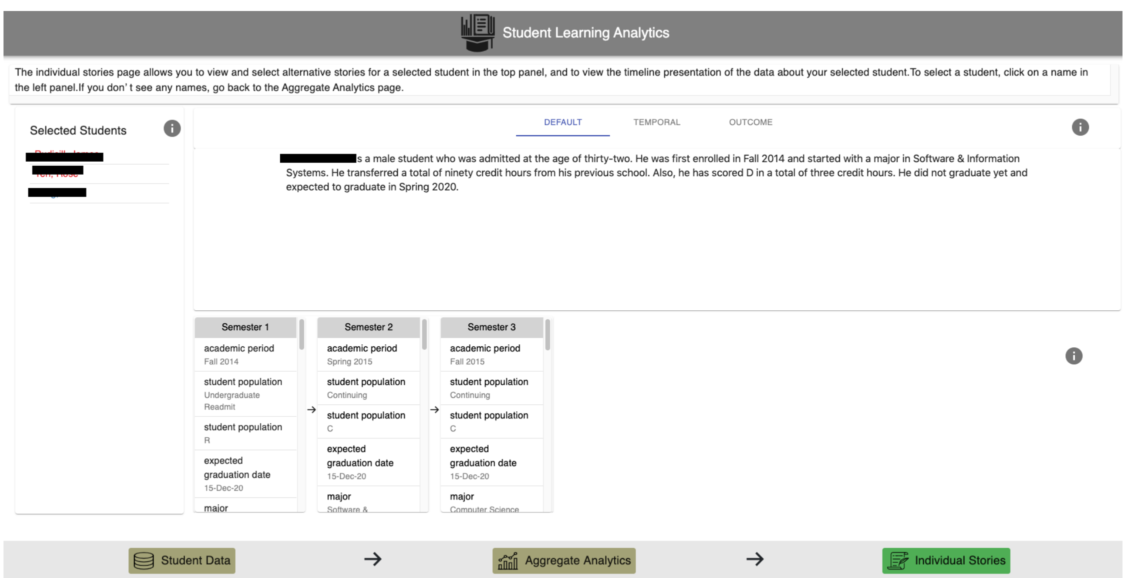


Figure A.3: Individual Stories of Selected Students

The bottom right part of the page shows progress: a timeline that shows the background and outcome node and all semester nodes for that student. Each node contains the values of the selected features in the student data model and some additional details for each semester.

A.3 Force Plots of SHAP Analysis on Bidirectional LSTM

We include the force plots to demonstrate the effect of percentages of C, D, and F we discussed in section 6.2.2.1.

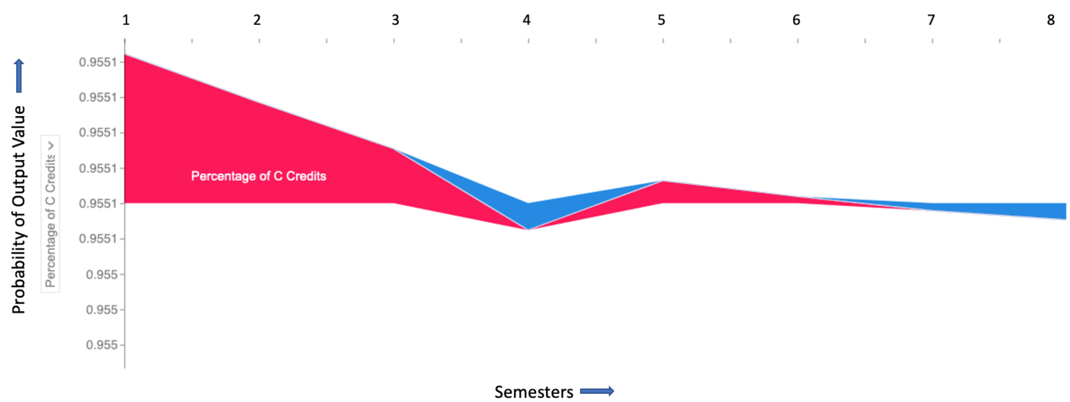


Figure A.4: Effect of "Percentage of C credits" on the value of CGPA

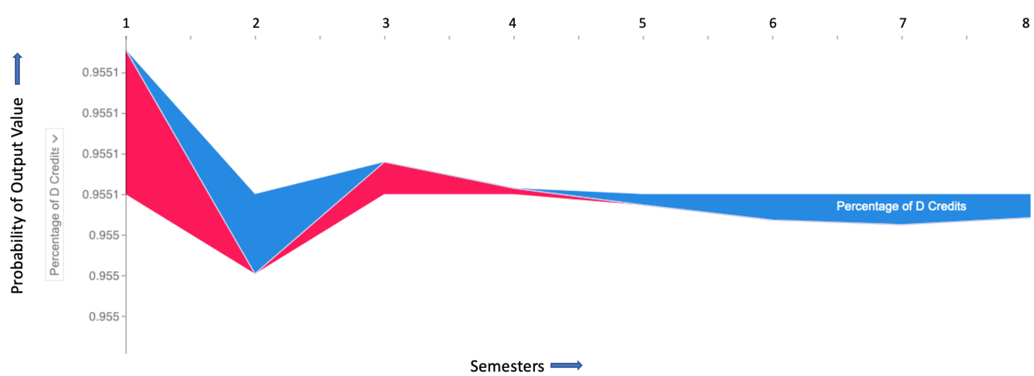


Figure A.5: Effect of "Percentage of D credits" on the value of CGPA

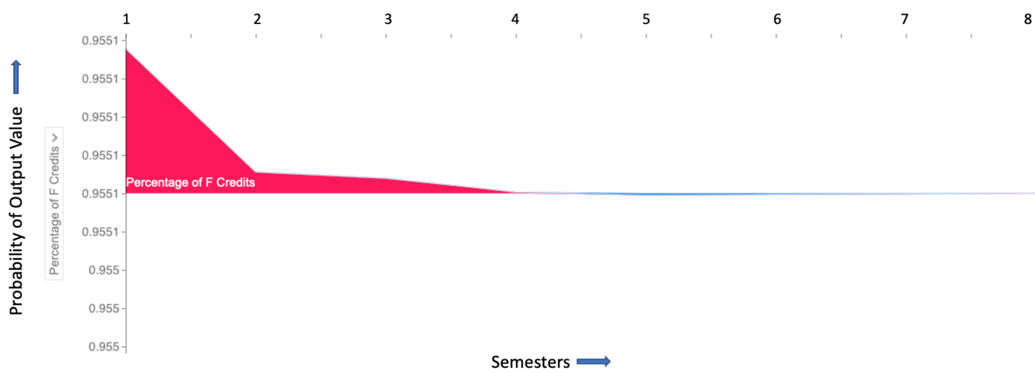


Figure A.6: Effect of "Percentage of F credits" on the value of CGPA