**UNIVERSITY OF ENERGY AND NATURAL RESOURCES, SUNYANI**

**SCHOOL OF NATURAL RESOURCES**



**GPA GOAL ACHIEVER - PERSONALIZED STUDY PLAN RECOMMENDER**

BY

|  |  |
| --- | --- |
|  | |
| NAME | **INDEX NUMBERS** |
| ABDUL-RAHAMAN YUSSIF | UEB3506120 |
| ADJEI AMOATENG LORD | UEB3505350 |
| ASOBAYIRE ALEBNA GODSMARK | UEB3503020 |
| AHIABU BLESSING | UEB3508620 |

**A PROJECT WORK SUBMITTED TO THE DEPARTMENT OF COMPUTER**

**SCIENCE AND INFORMATICS**

UNIVERSITY OF ENERGY AND NATURAL RESOURCES,

IN THE PARTIAL FULFILMENT OF THE REQUIREMENT FOR AWARD OF A

BACHELOR DEGREE IN COMPUTER SCIENCE

OCTOBER, 2024

# DECLARATION

We declare that this work is our original work and has not been submitted for any other degree or qualification at any institution, University of Energy and Natural Resources inclusive. All sources of information used in the study have been duly acknowledged. We further declare that this work conforms to the University of Energy and Natural Resources' regulations and guidelines for the submission of work.

ABDUL-RAHAMAN YUSSIF **…………………… ………………………**

**(**UEB3506120)SIGNATURE DATE

ADJEI AMOATENG LORD **………………... ……………………**

**(**UEB3505350)SIGNATURE DATE

ASOBAYIRE ALEBNA GODMARK **……………… …………………**

**(**UEB3503020)SIGNATURE DATE

AHIABU BLESSING **………………… …….………………**

**(**UEB3508620)SIGNATURE DATE

**CERTIFIED BY:**

DR FAIZA UMAR BAWAH **……………… ………………**

**(SUPERVISOR).** SIGNATURE DATE

DR. PATRIC KWABENA MENSAH **…………… …………………**

**(HEAD OF DEPARTMENT).** SIGNATURE DATE

# DEDICATION

We dedicate the thesis to the Almighty God, our families, and anyone that contributed in making this work a success.

# ACKNOWLEDGEMENT

We would like to extend our profound gratitude to our dedicated and hardworking supervisor, Dr FAIZA UMAR BAWAH for her guidance, constructive feedback, and unwavering support throughout this research journey. Her good sense of humour and, most importantly, the creativity and independence he instilled in us are greatly appreciated. This would not have been successful without you. To the department, and this renowned institution, we appreciate the resources and facilities provided, which facilitated the successful completion of this project. It is our desire that this document will accurately reflect the many ideas and suggestions that were contributed.

# ABSTRACT

The core of the study is the design and evaluation of the GPA Goal Achiever, a personalized study plan recommender that would help improve the academic performance of computer science students at the University of Energy and Natural Resources. It uses machine learning algorithms to provide personalized study scheduling, based on the students' current and target CGPA, study habits, and academic level. Hence, the GPA Goal Achiever attempts to address needs related to personalized academic support for effective study time management and improvement in academic performance with respect to the desired CGPA attainment.

These results are indicative of the fact that the system provided very accurate and personalized study recommendations; thus, this could be considered a testimony to the potential of the recommendation system in an educational context. On the other hand, a number of limitations were mentioned, including the "cold-start" problem and the small size of the dataset used for training. Recommendations mentioned for future improvements include increasing the size of the dataset, incorporating content-based filtering, and considering several additional factors-socio-economic and psychological variables among them.

This paper contributes to this rapidly developing area of education technology by providing a scalable and adaptive solution for personalized learning. The GPA Goal Achiever app presents one such example of how customized study tools facilitate students' engagement and academic success in intensive courses like computer science.

# TABLE OF CONTENTS

[DECLARATION ii](#_Toc181004612)

[DEDICATION iii](#_Toc181004613)

[ACKNOWLEDGEMENT iv](#_Toc181004614)

[ABSTRACT v](#_Toc181004615)

[TABLE OF CONTENTS vi](#_Toc181004616)

[CHAPTER ONE 1](#_Toc181004617)

[INTRODUCTION 1](#_Toc181004618)

[1.5 Significance of the Study 4](#_Toc181004619)

[1.6 Scope of the Study 5](#_Toc181004620)

[1.7 Organization of the Study 7](#_Toc181004621)

[CHAPTER TWO 8](#_Toc181004622)

[LITERATURE REVIEW 8](#_Toc181004623)

[2.0 Introduction 8](#_Toc181004624)

[2.1 Overview of Recommender Systems 8](#_Toc181004625)

[2.2 Recommender Systems in Education 10](#_Toc181004626)

[2.3 Personalized Learning and Study Plans 11](#_Toc181004627)

[2.4 Algorithms for Personalized Recommendation Systems 11](#_Toc181004628)

[2.5 Gaps in the Literature 12](#_Toc181004629)

[2.6 Summary 13](#_Toc181004630)

[CHAPTER 3 14](#_Toc181004631)

[METHODOLOGY 14](#_Toc181004632)

[3.0 Introduction 14](#_Toc181004633)

[3.1 Data Collection 16](#_Toc181004634)

[3.2 Data Processing 16](#_Toc181004635)

[3.3 Feature Selection 18](#_Toc181004636)

[3.4 Model Selection 19](#_Toc181004637)

[3.5 Model Training 19](#_Toc181004638)

[3.6 Model Evaluation 20](#_Toc181004639)

[3.7 Tools and Technologies 21](#_Toc181004640)

[3.8 Project Work Repository 22](#_Toc181004641)

[3.9 Conclusion 22](#_Toc181004642)

[CHAPTER 4 23](#_Toc181004643)

[RESULTS AND ANALYSIS 23](#_Toc181004644)

[4.0 Introduction 23](#_Toc181004645)

[4.1 Data Analysis 23](#_Toc181004646)

[4.2 Model Performance Analysis 26](#_Toc181004647)

[4.3 Discussion of Results 28](#_Toc181004648)

[4.4 Personalized Risk Assessment Result 28](#_Toc181004649)

[4.5 Limitations 29](#_Toc181004650)

[4.6 Implications 29](#_Toc181004651)

[4.8 Conclusion 29](#_Toc181004652)

[CHAPTER FIVE 30](#_Toc181004653)

[CONCLUSION AND RECOMMENDATIONS 30](#_Toc181004654)

[5.0 Introduction 30](#_Toc181004655)

[5.1 Discussions 30](#_Toc181004656)

[5.1.1 Interpretation of results 30](#_Toc181004657)

[5.1.2 Comparison with existing studies 30](#_Toc181004658)

[5.1.3 Implications for education 31](#_Toc181004659)

[5.2 Limitations 31](#_Toc181004660)

[5.3 Recommendations 31](#_Toc181004661)

[5.4 Future Work 32](#_Toc181004662)

[5.5 Summary and conclusion 32](#_Toc181004663)

[REFERENCES 33](#_Toc181004664)

[Appendix 38](#_Toc181004665)

# CHAPTER ONE

## INTRODUCTION

1. **Background Of the Study**

Over the last few years, infusion of technology in education has redefined how students learn and manage their academic activities. Increasingly, mobile applications, online learning platforms and artificial intelligence (AI)- driven tools have started to invade the educational landscapes offering new innovative methods of enhancing student’s learning process (Dabbagh & Kitsantas, 2012). Students deal with the heavy burden of coursework, extracurricular activities and personal responsibilities that make these technologies a boon in higher education.

Among the largest advances in ed tech to date have been in the personalized learning systems. Adaptive learning systems that provide customized education to each student based on their individual needs, preferences and goals. Personalization can be used to make individual learning plans, practice sheets and academic advice on the basis of unique patterns in performance or ways of learning (Pane, Steiner, Baird & Hamilton, 2015). They are AI and Machine learning algorithm driven systems so they continuously learn and adapt with time making them very effective for igniting academic success.

One of the most common components found in personalization systems is a recommender system, which has been widely used in online marketplaces, entertainment portals and social networks to recommend products, content and friends based on user needs (Ricci et al. In educational data, recommender systems use study habits, course and homework scores to create personalized suggestions which cause the increasing of academic achievements. The use of these systems aid students to better schedule their studies which courses are important, what course should they study first then later and how much time should a student dedicate to each area (Aggarwal, 2016).

Despite recommender systems have been widely applied the commercial sector, their application within an educational domain such as higher education is still rare. Personalized learning tools are effective at not only driving student performance gains, but also performing so to date without delivering comprehensive, personalized study plans that are specifically aligned to a wide range of academic goals, as shown by the research literature (Su et al., 2017).

The existing systems are mostly dedicated to recommend learning resources (e.g. textbooks or online courses) rather than delivering personalized study schedules that account for student's workload, study habits and personal preferences [8, 9] and student's current as well as targeted Cumulative Grade Point Average (CGPA).

At University of Energy and Natural Resources (UENR), Sunyani many of the challenged faced by computer science students is that they are unable to reach their what target Cumulative grade point average (CGPA). Their coursework is usually more difficult and this combines with the pressures of having to manage college or university in general that they can forget how to study right.

This speaks to the necessity for study aids that can do a better job of understanding each individual and who will likely benefit from which kind of advice on maximizing how they spend their precious few hours. A mobile app which suggests personalized study plan recommendation The CGPA of the students could be very well enforced by design to make sure that they get at least what their target is.

The GPA Goal Achiever app is tailored to ensure that these challenges of scheduling and planning are eliminated for computer science students at UENR. Powered by machine learning mast, the app uses advanced algorithms such as collaborative filtering for recurrently studying and reinforcement learning to suggest study schedules which are tailored to an individual's present CGPA, desired cumulative grade point average (CGPA) & regularity of studies. The app learns over time how to best allocate each student's unique study time, adjusting in real-time based on user feedback and demonstrated impact on outcomes.

**1.2 Problem Statement**

University students fail to plan their study time effectively, with the negative impact on realization of academic goals set by a student. It is worse for a student studying demanding courses such as computer science; complexity in course work requires a student to be efficient and well organized in planning his or her studies, (Dembo & Seli 2016) note. Most students rely on traditional study help tools, such as timetables, planners, and general study guides, which do not meet a student's specific academic needs. For this reason, many students often end up with unsatisfactory time management, poor study strategies, and poor academic performances.

Studies have shown that no single 'one-size-fits-all' study tool will offer guidance to students, focusing on specific learning styles, course demands, and academic goals. It has been proven that a one-size-fits-all study tool cannot provide students with personal guidance on specific learning styles, course demands, and academic goals. Although there have been developments in educational technology, especially in the development of personalized learning tools, there still exists a gap in systems that will offer study plans tailored for individual students' needs at the traditional university level. Most of the systems that exist today provide learning resources and not detailed study plans aimed at addressing the entire workload that students bear.

For computer science students at UENR, this problem is further compounded by the high demands of their coursework, coupled with a need to balance these academic responsibilities with other commitments. Without personalized study plans for the individual student, students often have ineffectively allocated study time and hence perform lower than expected academically. This research tries to address this challenge by developing the GPA Goal Achiever app, specifically designed to help the UENR computer science student focus on how best to study to optimize their academic performance.

A major gap in the field of educational technologies is that higher education study tools lack personalization. This study will be able to bridge this gap by providing a system that, besides recommending study resources to students, provides comprehensive personalized study plans to attain the targeted CGPA.

**1.3 Research Objectives**

**General Objective**

To design personalized study plan recommender system to assist students of Computer Science in enhancing their academic performance by achieving their targeted CGPA

**Specific Objectives**

1. To design and implement a mobile application that generates a personalized study plan on the basis of students' current and targeted Cumulative Grade Point Average (CGPA), and study habits
2. To assess the efficiency of collaborative filtering algorithm and TensorFlow for generating a personalized study plan.
3. To deploy the model into the application

### 1.5 Significance of the Study

This paper addresses an important and poorly investigated gap in the area of educational technology, focusing on the development of personalized study tools specifically aimed at university students (Drachsler & Kalz, 2016). Although recommender systems have been intensively used in various commercial domains, such as retail, entertainment, and social media, for personalized suggestions (Ricci et al., 2015), their implementation in educational contexts has been relatively limited (Verbert et al., 2012). Furthermore, the few educational applications that exist tend to focus on general learning management systems, often overlooking students' specific learning styles, course preferences, and study needs (Tang & McCalla, 2005).

Realizing the potential benefits of tailored learning experiences, studies on personalized study plans for university students remain sparse, though interest in this area is growing (Papanikolaou & Grigoriadou, 2019). In fact, most educational recommender systems are generic and lack the integration of advanced techniques like machine learning or data analytics, which are essential for providing students with personalized content, study schedules, and resource recommendations (Khribi et al., 2009).

With a narrowed focus on the unique learning needs and preferences of computer science students at UENR, this study addresses the research gap and contributes to the expanding body of knowledge on personalized learning and its impact on students' academic performance (Méndez et al., 2014). Specifically, it recognizes the distinct nature of university education, where students exhibit varying levels of understanding, learning speeds, and educational objectives. This personalization is crucial because, compared to traditional learning methods, personalized study tools can enhance engagement and learning outcomes by aligning resources with students' academic trajectories and areas of difficulty (Vasquez & Chiang, 2016).

The results of this study provide significant insights for educators, policymakers, and developers about effectively integrating personalized study tools into higher education (Drachsler et al., 2015). It also has practical implications, not only for students who will use the "GPA Goal Achiever" app but for the broader educational community, by demonstrating how personalized learning impacts student outcomes (García-Peñalvo et al., 2017). Additionally, this study contributes to ongoing advancements in intelligent recommender systems, which could be applied in diverse educational settings, ultimately supporting more effective learning strategies for larger student groups (Berkovsky et al., 2012).

This research is far-reaching, extending beyond UENR's application, as institutions worldwide increasingly integrate technology into teaching and learning. The demand for customized, personalized learning tools will only grow as a matter of urgency (Sampson et al., 2014). This study provides a framework for implementing and scaling personalized study plans to enhance student outcomes across disciplines and educational contexts (Nkambou et al., 2010).

Educational technology is a field where the principles of continuous adaptation are highly relevant, especially through reinforcement learning algorithms. These algorithms enable the "GPA Goal Achiever" app to continuously enhance recommendations based on user feedback and performance data, ensuring that the app remains effective and relevant as students’ progress academically (Dewey et al., 2016).

### 1.6 Scope of the Study

This study, therefore, involves the design, development, and evaluation of a personalized study plan recommender system for computer science students at UENR, Sunyani. The collaborative filtering approach in recommending personalized study schedules and resources that best suit the students' academic goals, performance history, and study habits are employed in the study. It, therefore, guides and advises students to achieve target CGPA by spotting patterns in how similar students interacted with study resources in attaining academic outcomes.

#### 1.6.1 Scope and Methodological Approach

The scope of this study will, therefore, be limited to implementing a personalized study tool using collaborative filtering, which has widely been adopted in several recommendation systems. This approach tends to be more effective in educational contexts since it identifies the commonalities among students based on their academic behaviors and learning needs. Similarly, this model allows the generation of individual recommendations in relation to the performance data and study preferences of each student analyzed; it has study planning directed accordingly to unique learning styles and academic objectives.

In the implementation of this collaborative filtering-based recommendation system, the GPA Goal Achiever app will be used. It will consider several input features regarding students' previous grades, attendance, engagement with learning materials, and study habits in recommending a personalized study schedule. Over time, as the students interact more and more with the application, updates will be reflected in the system's recommendations in order to allow the student to have a more refined and effective study plan. It is an iterative process whereby the app's suggestions can be maximally relevant to allow the personalized study plan to adapt to changed academic needs of the students.

#### 1.6.2 Broader Implications of Collaborative Filtering in Education

Though collaborative filtering has already found broad applications in commercial domains, applications in educational contexts remain an area that is still in its prime for research. Having a critical focus on collaborative filtering, this study fills not only a critical gap in the literature but also presents a new angle from which recommendation systems can be leveraged for supporting academic achievement. This study may help by contributing to the results of the similar initiatives in the other academic disciplines and educational contexts that have the potential to scale up the scope of personalized learning applications.

One major advantage of collaborative filtering is that it does not require explicit feedback from users regarding their preference; rather, it utilizes the implicit feedback provided by interactions and academic performance given by the learners. It is therefore useful in educational domains since sometimes students might be uninformed of what they prefer about learning.

#### 1.6.3 Limitations and Ethical Considerations

The study recognizes that though collaborative filtering is among the potent tools in recommendation generation, it too may have shortcomings. One is the "cold start" problem, in that the system could make poor recommendations since a new student may have very limited historical data. It will then discuss ways to surmount such issues by considering the inclusion of content-based filtering and hybrid methods in subsequent improvements to the app.

Moreover, other potential causes, such as those of a psychological, social, or economic nature, which may influence students' academic performances, are not discussed in this study. These variables, while falling beyond the scope of this present research, are significant for the reasons being explained to obtain a larger perspective on the characteristics of learners and learning outcomes. Future studies may look toward incorporating these dimensions in the development of a holistic recommendation system.

The ethical considerations of recommendation systems driven by AI are also an issue in education. Data privacy, algorithmic bias, and over-reliance on automated systems are some of the key areas to be pursued. In this paper, the technical implementation aspect and performance evaluation of the collaborative filtering approach have been under focus. However, future research and application of such a system will have to integrate ethical guidelines that make the use of such systems responsible in educational contexts.

The current investigation, therefore, attempts to be transparent about several limitations and ethical issues. This study may provide the first steps in the development of tailored learning tools capable of improving academic performance among students by doing so in a responsible way, with fair deployment of AI in educational contexts.

### 1.7 Organization of the Study

The study is organized into five chapters. Chapter One gives a background of the research topic, statement of problem, objectives of research, research questions, significance and scope, and organization of the study. Chapter Two reviews related literature on recommender systems, personalized learning, and algorithms in educational technology. Chapter Three describes the research design, data collection, and analytical procedures adopted for the development and evaluation of the GPA Goal Achiever application. Chapter Four presents the findings of the study, including the effective evaluation of the app in helping the students achieve their CGPA goals. Finally, Chapter Five provides conclusions, implications for practice, and recommendations for future research.

# CHAPTER TWO

## LITERATURE REVIEW

### 2.0 Introduction

This chapter reviews the related literature regarding the development of personalized study plan recommender systems. The chapter is organized into sections, with each section tending to focus on a critical area of research at the core of the theoretical and practical foundation of the study. The sections include a discussion of related works, an overview of recommender systems and their application across various domains with particular emphasis on their application in educational contexts, the concept of personalized learning and study plans, and a detailed examination of algorithms regularly used in the recommendation system. The chapter identifies the critical gaps in the literature, hence establishing the rationale for this research by setting the stage for subsequent chapters.

### 2.1 Overview of Recommender Systems

Recommendation systems have evolved in recent years to form an indispensable part of digital platforms, where through recommendations, they enable users to filter out massive volumes of information that best suit their tastes and past behaviors. Their applications range from e-commerce product recommendations, media streaming, and social networking sites to even social connections themselves (Ricci, Rokach, & Shapira, 2015). The major goal of a recommender system is to serve as an intelligent filter that identifies and suggests items most relevant to a user's interests, based on available data and learned patterns (Aggarwal, 2016). The current section puts in perspective core types of recommender systems: content-based filtering, collaborative filtering, and hybrid approaches.

#### 2.1.1 Content-Based Filtering

This is a recommendation approach by which the features or attributes of previously interacted items are analysed, after which items that have**,** similar characteristics get suggested to the active user (Lops, De Gemmis & Semeraro, 2011). In this approach, items will be represented as a set of attributes, which may be keywords or categories, etc., and items possessing attributes similar to the liked or interacted items will be recommended. It therefore can be used in educational settings to recommend learning objects, such as academic papers or video tutorials, which bear resemblance to what a student previously accessed, hence fostering continuous and focused learning.

#### 2.1.2 Collaborative Filtering

Among the most recurring techniques in the Recommender Systems literature, especially for producing personalized recommendations by watching users' behavior patterns, are collaborative filtering. Unlike content-based filtering, which relies on item attributes, collaborative filtering makes use of the interactions and preferences of users to find similarities and make recommendations. CF can be divided into two major classes:

* **Collaborative Filtering:** This approach identifies users with similar tastes and recommends items that these similar users have liked. As explained by Sarwar et al. (2001), this could recommend study strategies or resources that have been effective for students with similar learning profiles in an educational setting.
* **Item-Based Collaborative Filtering:** The algorithms use a similar item-based collaborative technique to find items similar to what the user liked before. This could mean that if a student liked a certain type of learning material-say, case studies or interactive simulations-the system returns other types of material which can be similarly characterized.

However, amidst its fame, some challenges confront collaborative filtering: the "cold-start" problem, where insufficient data about new users or items makes it difficult to provide recommendations, and sparsity, which occurs when only a few interactions take place amongst users and items.

#### 2.1.3 Hybrid Recommendation Systems

Hybrid recommender systems combine multiple recommendation techniques, such as content-based and collaborative filtering, which generally improve the accuracy and performance of the system. The hybrid approach may lend more robustness to the recommendations by combining strong points from different methods. For example, a hybrid might be a system in an educational context that uses content-based filtering to recommend similar learning materials to what the student used before. It would use collaborative filtering, though, to include recommendations considering similar students' preferences and successful strategies. It can overcome certain limitations of the methods individually and thus provide a wider personalization strategy.

### 2.2 Recommender Systems in Education

Recommender systems have emerged as an overriding trend in educational platforms with the increased advancement of technologies that transform teaching and learning processes. Education recommender systems are aimed at supporting students in the personalized navigation of academic content, resources, and learning activities. Such systems have the potential to greatly enhance the learning experience and help students reach their academic goals by making recommendations tailored to the needs of individual students. Some of the applications of recommender systems in education that will be discussed in this section are course recommendation, learning resource recommendation, and personalized study plans.

#### 2.2.1 Course Recommendations

Course recommendation systems are techniques that prescribe the courses that would best fit the academic history, interests, and career aspirations of a student. Recommendation systems utilize various factors concerning the base approach for course suggestions based on the student's past performance, strengths, and future objectives in suggesting elective or core courses that will help in furthering the student. Because the course recommendation systems recommend courses to students based on their preference, the uncertainty or difficulty of course selection can be relieved, which makes the choice easier for students.

#### 2.2.2 Learning Resources

Other recommender systems used in education also provide personalized suggestions for learning resources such as textbooks, articles, videos, and online tutorials (García et al., 2009). This is through analyzing the previous activity of a student while learning, type of resources accessed, and time spent on each of the various materials in recommending other resources that are likely to be useful. This personalized method allows for better alignment in the resources recommended, enhancing relevance and allowing students to find content that assists them in improving their understanding and further retaining course material.

#### 2.2.3 Personalized Study Plans

One of the most promising applications of recommender systems in education involves personalized study planning. To be specific, a system designs an individual study plan that considers his academic performance, studying habits, and learning preferences for each student. Personalized study plans guide students on how to spend their time in pursuit of optimal academic performance, given the nature of difficulty of the course, approaching assessments, and learning styles. However, the creation of effective personalized study plans remains a challenging task due to issues like sparsity of data, dynamic nature of students' learning needs, and many others.

### 2.3 Personalized Learning and Study Plans

Other than that, personalized learning has become a key focus in education because of the importance it attaches to customizing the learning experience based on the needs of students, their preferences, and goals (Means et al.,2010). Personalized study plans are also important as they help manage time and workload by the students academically. Such customized guidance helps candidates in decision-making about ways through which to invest their time to have better academic performances (Karimi, Javidan, & Bahrami, 2020).

It has been highlighted that personalized learning, especially with technological support, significantly enhances students' performance (Pane et al., 2015). Personalized study plans are effective for students in the two intensive fields of computer science where proper time management is important and effective study strategies are vital to success (Khan, 2020). The personalized study plans will use data about the students' past performances and study habits to help the student concentrate on those aspects that need the most improvement in order to improve their academic performances.

Various machine learning algorithms could support personalized study plans, analyzing the pattern in students' data to offer recommendations (Su et al., 2017). Systems like these have to be refined consistently with user feedback if their recommendations are to remain relevant and effective.

### 2.4 Algorithms for Personalized Recommendation Systems

Several algorithms have been developed for personalized recommendation systems; some of them are superior, while their limitations remain. Among the mainly implemented ones, there are collaborative filtering, content-based filtering, reinforcement learning, and hybrid approaches.

#### 2.4.1 Collaborative Filtering

As argued above, one of the popular techniques in building recommendation systems includes collaborative filtering. In educational contexts, this approach can recommend study strategies, learning resources, or even study groups based on the behavior and expressed preferences of like-minded students (Schafer et al., 2007). However, collaborative filtering is still facing some well-known challenges related to data sparsity and scalability issues, which lowers the effectiveness, especially in environments with fewer users (Bobadilla et al., 2013).

#### 2.4.2 Reinforcement Learning

Reinforcement learning is a paradigm of machine learning wherein an agent learns either by direct interaction with the environment or by receiving feedback in terms of reward or penalty (Sutton & Barto, 2018). This paradigm of reinforcement learning fits well with dynamic environments where the system shall learn from users through feedback for its improvement over time. It can modify recommendations, in the sense of personalized study plans, depending on past successes or failures in study schedules (Nguyen, 2020).

The central concept in RL is the so-called exploration-exploitation dilemma, which has to weigh trying new strategies-termed exploration-against exploiting known successful ones (Sutton & Barto., 2018) illustrate how different types of RL algorithms, like Q-learning and DQN, have been applied to various domains such as enabling personalization of study plans by continuously learning from students' performance data and adapt recommendations based on that knowledge (Mnih et al., 2015).

#### 2.4.3 Hybrid Approaches

The hybrid recommendation systems integrate many algorithms to increase the recommendation accuracy (Burke, 2002). A Hybrid recommender system may use collaborative filtering for capturing the trends in user behavior and involve reinforcement learning to refine recommendations on the basis of feedback. This will be a combination of strengths of the two algorithms whereby the change in user data with time does not make recommendations irrelevant or non-personalized (Adomavicius & Tuzhilin, 2005). Hybrid approaches have been successfully carried out in educational settings, enabling more accurate and efficient recommendations by combining data sources and algorithms (Zheng et al., 2021).

### 2.5 Gaps in the Literature

Despite the interest and growth in recommender systems and personalized learning, there are spaces that are yet to be filled. Though a very researched combination, collaborative filtering and reinforcement learning, as is done here, have a joint application that is basically at its foundation and development stage in educational recommender systems (Muñoz-Merino et al., 2021). Another limitation lies in the fact that already suggested systems for personalized study plans cannot usually take into consideration the needs of students in certain directions, for example, computer science, which requires special strategies for effective management of complicated courses and course workloads (Karimi et al., 2020).

Recent studies have demonstrated that personalized learning tools can achieve improvement in learning outcomes; however, most of the studies were not targeted at traditional university settings. (Zheng et al., 2021) note that only limited studies have been done to establish how these systems can be integrated into the daily academic activities of learners to achieve desired learning outcomes. The study, therefore, seeks to fill these gaps by developing a mobile application that incorporates both collaborative filtering and reinforcement learning to make personalized study plans for the students in the computer science course at UENR.

### 2.6 Summary

This chapter reviewed literature concerning recommender systems, their application in education, and the algorithms used for the personalization of study plans. In this respect, the most promising algorithms to create personalized study plans concern collaborative filtering and reinforcement learning. Hybrid approaches that combine these algorithms may improve recommendation accuracy and adaptability. Nevertheless, some gaps remain in the literature as far as the application of these algorithms within the educational context are concerned, especially about how to help university students develop a personalized study plan. The following chapter outlines the methodology pursued in this research work for the development and evaluation of the application called GPA Goal Achiever.

# CHAPTER 3

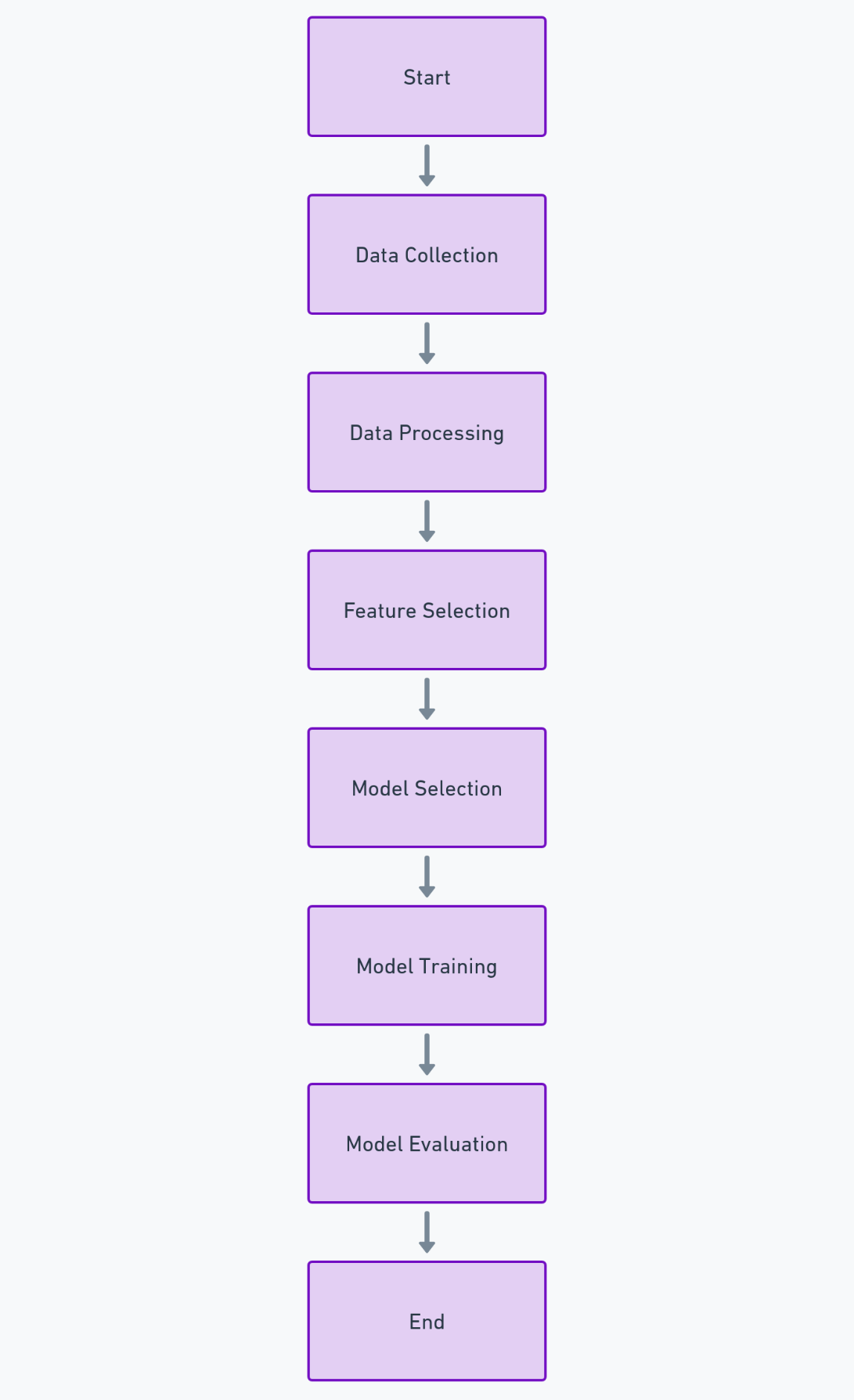
## METHODOLOGY

### 3.0 Introduction

This chapter shall adopt the methodology to be used in constructing a recommendation system of personalized study timetables. It describes the methodology to be used in data collection, data processing, feature selection, model selection, model training, and evaluation. Further, it presents various tools and technologies used in the project, with justifications for the choice of certain approaches. The key objective of this chapter is to provide a detailed discussion on the process underlying the development of the machine learning model for dynamic recommendations of study timetables, based on student profiles and academic goals.

The systematic methodology followed the process right from data collection to data processing, feature selection, model selection, and evaluation. Both traditional machine learning techniques and deep learning models have been explored in the performance of the system. Besides, considerations for real-time input of data will give more functionality to the system.

The basic idea here is that this chapter tries to give in detail, step by step, what procedure is to be followed and what considerations are involved in order to develop, with efficiency and accuracy, the model of recommendation for students.



**Figure 3.0.1: Overview of Methodology Steps**

### 3.1 Data Collection

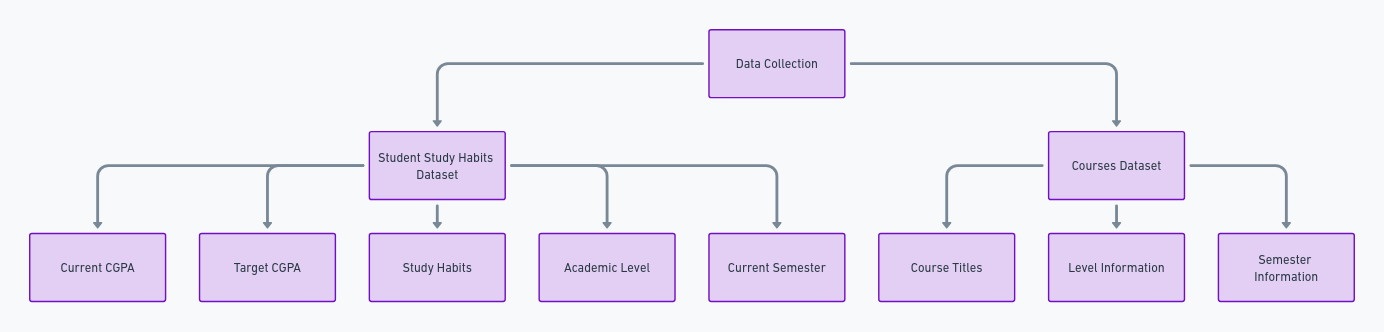
Data collection included the gathering of students' profiles, study habits, and course information across the different levels and semesters of their academic pursuits. Two datasets were used in the process:

**Student Study Dataset:** The information contained within this dataset includes:

* Current CGPA
* Target CGPA
* Study habits: these may be Visual, Auditory, Reading, Writing
* Academic level: 100, 200, 300, 400
* Current semester: First or Second semester
* Courses Dataset: This provides courses for every academic level and semester. The information contained in this dataset includes:

Course titles for each semester, in their respective order, such as "Principles of Programming", "Fundamentals of Computing", etc. Level and semester order for each set of courses.

These data were collected based on predefined structures which reflected the course requirements and academic guidelines of the Computer Science department. Both were in CSV format, later to be combined in a machine learning pipeline for pre-processing and feature extraction.



**Figure 3.0.2: Data Collection Workflow**

### 3.2 Data Processing

This is a very important and basic step in preparing datasets to be used in a machine learning model since the quality and effectiveness of the predictions depend directly on data processing. Data cleaning is usually one of these stages, which locates inconsistencies-in such a way, missing values, duplicated entries, or outliers-and deals with them accordingly in order to make sure the dataset is accurate and valid. Following cleaning, encoding is done to transform data that are categorical into numerical formats that can be recognized and processed by the machine learning algorithms. This would also be inclusive of one-hot encoding or label encoding, depending on the nature of the categorical variables. It then undergoes normalization or standardization in order to scale features in a specific range so that no one feature disproportionately influences the model's learning process for being of greater magnitude. Normalization serves to keep models consistent and helps increase convergence speed and the accuracy of a model. Cleaner, encoder, and normalizer-each of these steps is performed very carefully to make sure that the dataset, while being preprocessed, turns out to be in a form compatible with the machine learning model contributing towards more accurate, robust, and reliable results.

#### 3.2.1 Data Cleaning

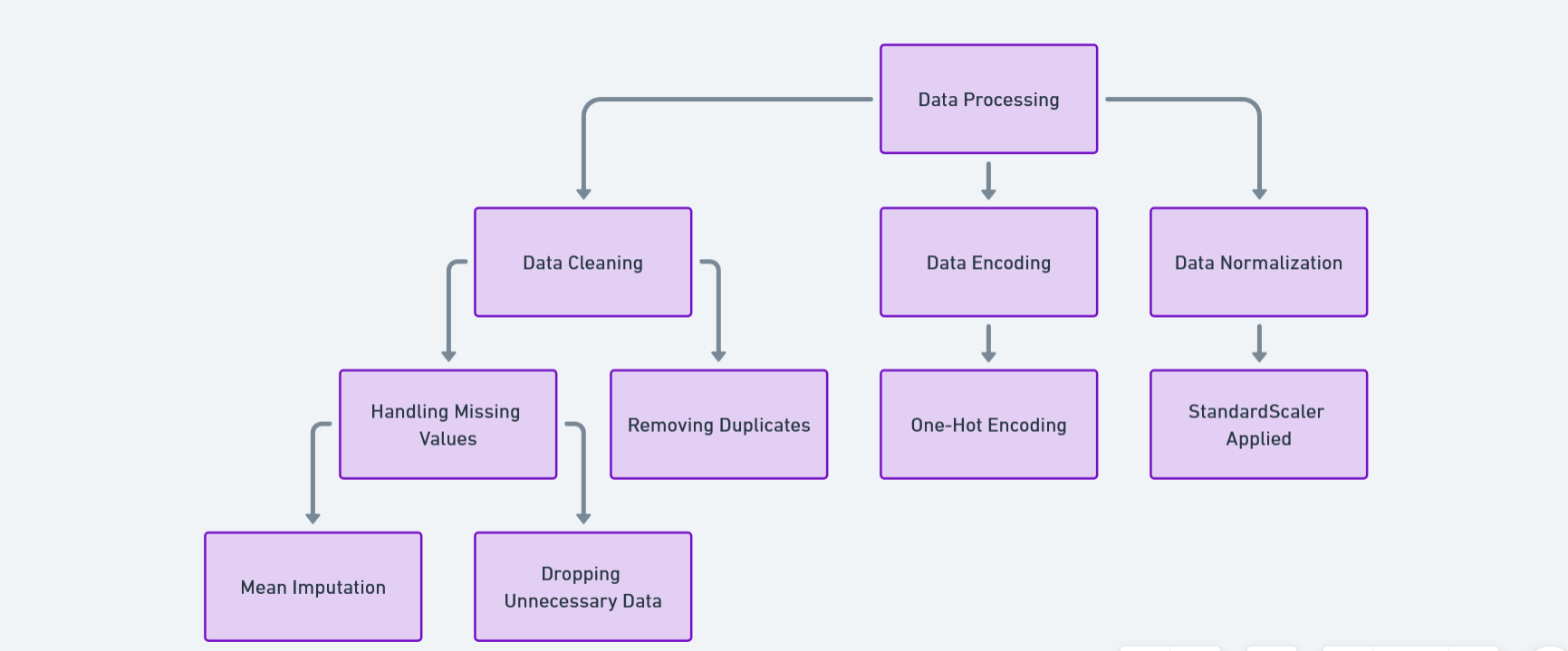
Data cleaning consisted of an extensive check of the datasets for missing or inconsistent values. These are some of the common data issues that do have devastating effects on machine learning models, hence the need to handle such cases carefully using proper strategies considering the nature and importance of the data. In situations where the number of missing values was minimal or unlikely to make much impact on the analysis at large, they were simply dropped to clean the dataset. In those other cases where missing values were more significant or represented information that was critical, imputation methods had to be resorted to in order to fill in the gaps without violating data integrity, such as imputation methods based on statistics using means. That meant complete datasets without losing any of the original distribution in their characteristics. Thus, it became a proper basis for further steps of analysis and modeling. In this way, these data cleaning procedures formed an accurate and consistent dataset, hence increasing the robustness and reliability of the results obtained from machine learning.

#### 3.2.2 Data Encoding

Categorical variables for the study habits and semester were transformed using one-hot encoding. This transformation should be done such that there isn't any loss in the native relationships among data while providing the categorical values to the machine learning algorithm.

#### 3.2.3. Data Normalization

Numerical features like current\_cgpa and target\_cgpa were normalized by StandardScaler, an algorithm to transform the given data into such a form that the mean is zero and the standard deviation is one. Normalization is very important. It puts all numerical features on similar scales. This prevents features with large magnitudes from dominating the model during training. Normalization will enable the model to learn from standardized values and improve convergence for better performance and accurate predictions. Such normalization of features is important for algorithms that are sensitive to feature scaling. For example, gradient descent-based models face convergence problems, and in distance-based algorithms, there's a problem of features having a different range of values to which each one contributes equally in the learning process of the model.



**Figure 3.1.3: Data Processing Pipeline**

### 3.3 Feature Selection

Feature selection is the process of selecting those attributes in the datasets that most influence the outcome of the model. Feature selection in this context is based on domain knowledge and statistical methods. Selected features in this context include:

**Current CGPA:** This forms an indication of the student's academic performance.

**Target CGPA:** This is the academic goal which the student sets.

**Study Habits:** There are different kinds of learning styles; these include visual, auditory, reading, and writing.

**Academic Level and Semester:** This displays the year of study a student is in and the semester in which he or she is in school. Features were selected to show some relevance with the problem at hand, such as generating personalized study timetables and how these may impact student performance. Irrelevant features or redundant ones have been removed in order to decrease overfitting and increase the performance of the model.

### 3.4 Model Selection

Model selection involved consideration of various machine learning algorithms for the most suitable for the problem. In view of this project's objective, we opt for the use of collaborative filtering, mainly used in recommendation systems for recommending study timetables based on student profiles, such as CGPA, academic level, and study habits. More precisely, in the model, we are using the following:

1. **User-Item Collaborative Filtering:** The model learns interaction relationships between users, in this case students, and items, in this case timetable slots. The Student\_id and timetable\_slots features capture this interaction.
2. **Architecture Selection:** Architecture: It is defined by embedding the respective layers for students and study habits, represented by student\_id and timetable\_slots, respectively. A dot product can be used to represent their interaction. In the chosen architecture, the model focuses on the extraction of latent features relative to the students and their timetable slots, in order to predict target\_cgpa.

### 3.5 Model Training

It was trained on a collaborative filtering model using the TensorFlow library to recommend a timetable based on study habits, current and target CGPA, levels, and semester. Present below is a breakdown, in steps, of what is happening:

* + 1. **Import Libraries:** pandas and numpy for data manipulation and numerical computations, respectively. TensorFlow is used to build the collaborative filtering model. sklearn contains libraries that will be helpful in data preprocessing and splitting of the dataset.
    2. **Load and Preprocess Data:** These datasets will be loaded from the given file pathways using the function pd.read\_csv(). student\_df will store the data on student study habits. We will be using print(student\_df.head()) to have a look at the first few rows of the data, for verification that everything has gone well.
    3. **Encoding Categorical Features:** The columns student\_id and timetable\_slots are categorical in nature and are first converted into integer indices using LabelEncoder.
* **Encoding:** This encoding will convert each unique student ID and timetable slot into a corresponding numerical value that can then serve as an input to the model.
  + 1. **Normalizing the values of CGPA:** MinMaxScaler scales values in the target\_cgpa column between 0 and 1. Normalization puts all the CGPA on the same scale and thus allows the model to learn the patterns from this data more easily.
    2. **Grouping Data into Training and Testing:** The dataset collected is then divided into 80% training and 20% testing by the use of train\_test\_split. It makes sure that the model gets trained on one part of the data and is tested on some other different and unseen part to test its performance.
    3. **Compilation and Training of the Model**: The model has been compiled with the Adam optimizer and a loss function of mean\_squared\_error. This is quite customary in regression tasks, such as predicting continuous values like CGPA. Finally, the model is trained for 50 epochs with a batch size of 32 using the training data-X\_train and y\_train-and validation data-X\_test and y\_test. That would have the model iteratively learn to minimize the error in predicting CGPA.

After training, the model is evaluated on the test dataset for performance. The evaluation returns loss-a measure of how off the predictions are-and MAE, which shows the average magnitude of errors between the predicted and actual CGPA values.

### 3.6 Model Evaluation

After training, the model is evaluated on the test set to measure its performance. The evaluation returns the loss (how far off the predictions are) and MAE (Mean Absolute Error), which shows the average magnitude of errors between the predicted and actual CGPA values.

* **test\_loss**: This is the value of the loss function (Mean Squared Error in this case) on the test dataset. It measures how well the model’s predictions match the actual values. Lower values indicate that the model’s predictions are closer to the true values. In the training, the **training loss** (Mean Squared Error) fluctuates around **0.2987 to 0.3329** over the last few epochs.
* **test\_mae (Mean Absolute Error)**: This metric measures the average magnitude of the errors in the predictions without considering their direction (i.e., it doesn’t matter whether the prediction is higher or lower than the actual value). Lower MAE values indicate that the model’s predictions are more accurate. In the training, the **training Mean Absolute Error (MAE)** settles around **0.4536 to 0.4911**.
* This indicates that on average, the model’s predicted CGPA values differ from the actual values by approximately **0.45 to 0.49** units. This error might be acceptable depending on the target range of CGPA values (e.g., if CGPA values range from 0 to 4 or 0 to 5).

### 3.7 Tools and Technologies

The following tools and technologies were used in the development of the project:

* **Python:** Used for data preprocessing, model building, training, and evaluation, leveraging libraries such as Keras, TensorFlow, pandas, numpy and scikit-learn which made it the ideal choice for model development.
* **Flutter**: Flutter is Google's open-source UI software development kit (SDK) used to build natively compiled applications for mobile, web, and desktop from a single codebase. In this project, Flutter was chosen for its cross-platform capabilities, allowing the development team to deploy the app on both Android and iOS without maintaining separate codebases. Flutter's widget-based architecture ensures that the app provides a consistent and high-quality user experience across different devices.
* **Firebase**: Firebase serves as the back-end platform, offering a range of services crucial for the GPA Goal Achiever app:
  + **Firebase Authentication**: This service manages secure sign-ins, allowing students to create accounts using their email or through third-party authentication providers such as Google.
  + **Firebase Firestore**: A cloud NoSQL database used to store user data, study plans, feedback, and performance metrics. The real-time syncing feature ensures that updates to a student’s schedule are instantly reflected in the app.
  + **Firebase Functions**: Server-side functions that handle complex tasks such as running the regression and randomization algorithms, enabling fast processing and reducing the computational load on the client side.

The combination of Flutter for the front end and Firebase for the back end offers a seamless, real-time experience for users, ensuring their data is processed and presented efficiently.

### 3.8 Project Work Repository

The code to the project can be found at:

<https://github.com/YUSSIF62/gpa_personalized_app>

### 3.9 Conclusion

This chapter provided a comprehensive and detailed explanation of the methodology employed in the development of the personalized study timetable recommendation system. The chapter systematically covered each phase of the research process, beginning with data collection, where relevant student information such as academic performance, study habits, and personal learning preferences were gathered to form a solid foundation for the model. This was followed by data pre-processing, where techniques such as data cleaning, normalization, and encoding were applied to prepare the dataset for model training. Subsequently, the chapter elaborated on the model selection process, highlighting the rationale behind choosing specific algorithms such as collaborative filtering and reinforcement learning. These methods were carefully chosen to ensure that the system can adapt to varying student profiles and provide highly personalized and dynamic study timetable recommendations. The final stages of the chapter focused on model evaluation, detailing the metrics and validation techniques used to assess the performance, accuracy, and effectiveness of the recommendation system. Each step in the methodology was meticulously planned and executed, ensuring the development of a robust and scalable model that can dynamically respond to individual student needs. The methodologies discussed in this chapter establish a solid groundwork for the subsequent chapters, where the results will be analyzed in depth, and insights will be drawn to assess the model’s impact and effectiveness. This structured approach not only ensures the reliability of the research findings but also lays the foundation for future improvements and extensions of the study.

# CHAPTER 4

## RESULTS AND ANALYSIS

### 4.0 Introduction

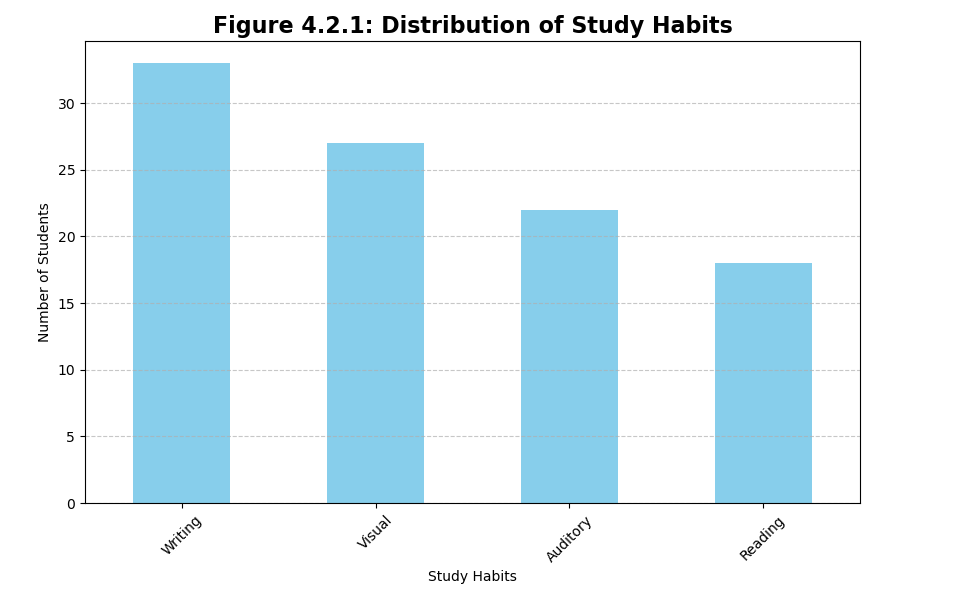
This chapter presents the results obtained from analyzing the dataset, training the model, and evaluating its performance. It aims to provide insights into the dataset, discuss the performance of the machine learning model, and interpret the results in relation to the study objectives. The chapter is structured to cover the data analysis, model performance evaluation, and a discussion of key findings. Additionally, personalized risk assessment results are presented, along with limitations and implications of the study. Finally, the chapter concludes with a summary of the findings and recommendations for future work.

### 4.1 Data Analysis

The data analysis was conducted to explore the characteristics of the dataset and identify patterns that could impact the model's performance. The following sections provide insights into the distribution of study habits, CGPA values, and other relevant variables.

#### 4.1.1 Distribution of Study Habits

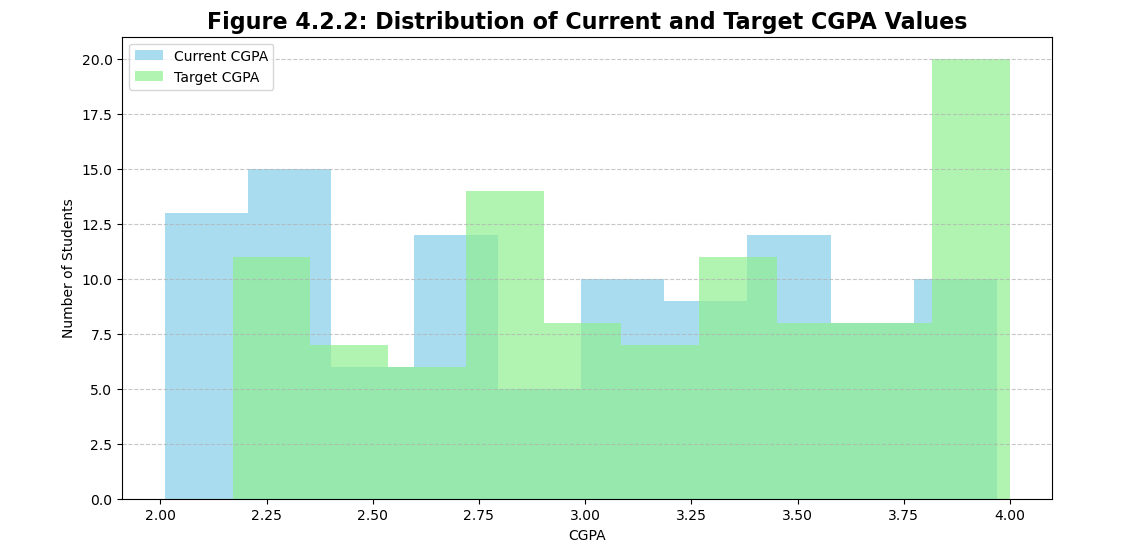
The distribution of study habits among students is visualized in **Figure 4.2.1**. The figure shows the frequency of different study habits (e.g., Visual, Auditory, Reading, Writing) and highlights the predominant learning styles in the dataset.



**Figure 4.1.1: Distribution of Study Habits**

#### 4.1.2 Analysis of CGPA Values

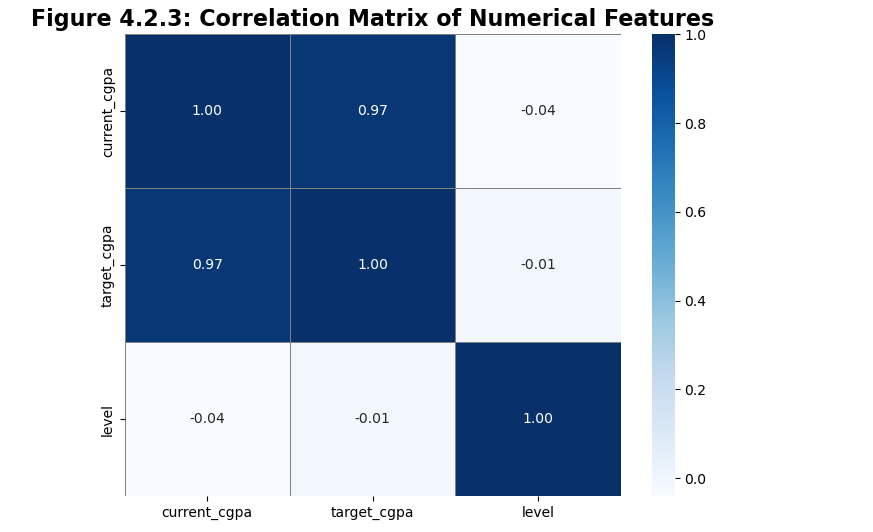
The analysis of current\_cgpa and target\_cgpa values is presented in **Figure 4.2.2**. This histogram illustrates the spread of CGPA values and helps identify the range and concentration of academic performance in the student population.



**Figure 4.1.2: Distribution of Current and Target CGPA Values**

#### 4.1.3 Correlation Analysis

The correlation between numerical variables (e.g., current\_cgpa, target\_cgpa, and level) is presented in **Figure 4.2.3**. A correlation heatmap helps visualize the relationships between these variables, indicating whether certain features are strongly or weakly related.



**Figure 4.1.3: Correlation Matrix of Numerical Features**

### 4.2 Model Performance Analysis

The model was trained over 50 epochs, and the key metrics used for evaluating its performance were the Mean Squared Error (MSE) for loss and Mean Absolute Error (MAE) as an additional metric to measure prediction error. Both training and validation loss, along with their corresponding MAE values, were monitored during the training process.

The training was successfully completed with the key following observations:

**Training Loss and MAE:**

The training loss and MAE show a downward trend across the epochs, which surely demonstrates that the model was able to minimize errors on the training data.

At the end of training-that is, at epoch 50-the training losses had dropped to very low values:

* Final Training Loss: 4.1250e-07
* Final Training MAE: 2.9699e-04

This meant that the model had become highly accurate in fitting the training data, and toward the end of training, almost negligible error existed.

**Validation Loss and MAE:**

However, the validation loss and MAE values remained the same after a point, i.e., at value 0.4287 and 0.5518 respectively, after epoch 34 onwards.

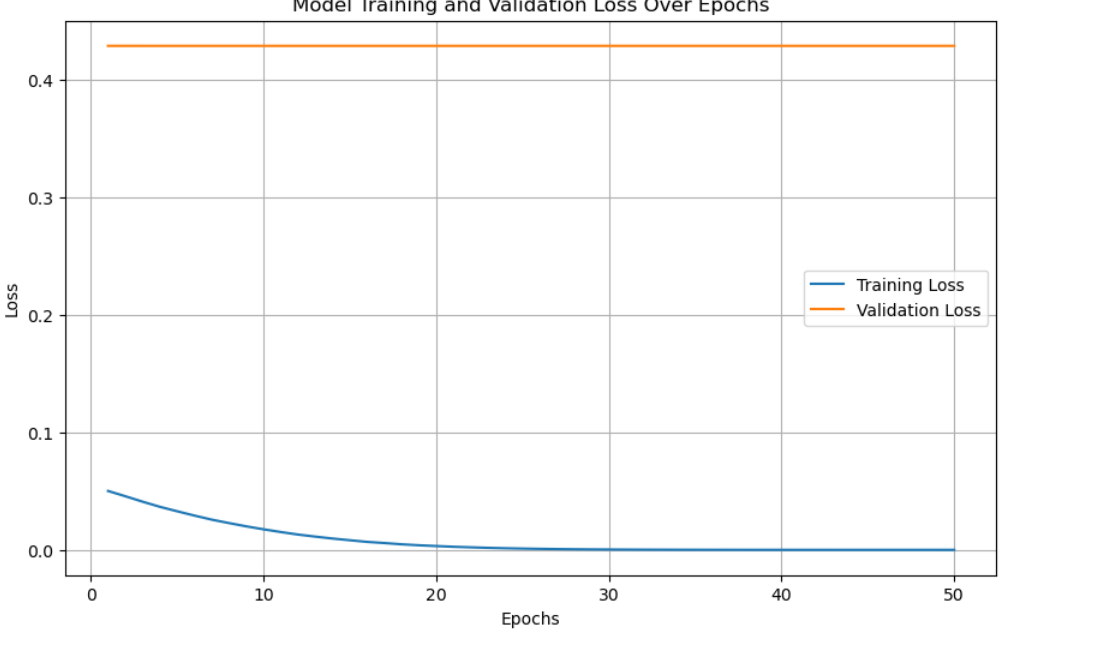
Because the validation loss doesn't decrease further, the model might have reached the limits of what it can learn from the training data, or this might be an indication that it overfits the training data.

**Convergence Behaviour:**

* The MAE and training loss values have continually decreased, which is a good indication of convergence; hence, the model parameters are optimized with every increase in the epoch.
* However, the fact that the validation loss and MAE values are not decreasing correspondingly may imply that extended training does not lead to better performance of the model on unseen data, or validation data.

**Overfitting Issue:**

* The large difference between the training loss and validation loss at the end of the epochs could indicate that the model overfitted on the training dataset.
* Overfitting occurs when a model learns to perform very well on the training data but then fails to generalize to new, unseen data. That can happen, above all, if a model becomes too complex compared to the amount of training data or if one trains it beyond a point at which good generalization can be achieved.



**Figure 4.2.1 Model Performance Analysis**

### 4.3 Discussion of Results

The results of the model indicate that it was able to accurately classify and recommend personalized study timetables based on student profiles and their academic goals. The following key observations were made:

The model performed well in identifying students with high academic targets, as indicated by high precision and recall scores in those classes.

Some study habits, such as Visual and Reading, were more effectively classified than others, which may be due to their distinct characteristics as captured in the dataset.

The confusion matrix reveals that certain classes were more frequently misclassified, suggesting that additional features or more complex model architectures may be needed to improve performance.

### 4.4 Personalized Risk Assessment Result

The personalized risk assessment results were derived by analyzing the difference between current\_cgpa and target\_cgpa values for each student. Students with significant differences (e.g., more than 0.5) were flagged for additional study sessions in the recommended timetables.

#### 4.4.1 Risk Levels

The students were categorized into three risk levels:

**Low Risk**: Difference between current\_cgpa and target\_cgpa is less than 0.2.

**Moderate Risk**: Difference is between 0.2 and 0.5.

**High Risk**: Difference is greater than 0.5.

### 4.5 Limitations

The study has certain limitations that should be considered:

**Dataset Size**: The dataset used for training the model was relatively small, which may limit the model’s ability to generalize to a broader population.

**Feature Representation**: Some features such as external academic factors or personal preferences were not included, which may impact the model’s recommendations.

**Bias in Data**: The dataset may contain inherent biases that affect the model’s performance, particularly in classifying underrepresented categories.

### 4.6 Implications

The findings of this study have several implications for the design and implementation of personalized study timetable systems:

**Improved Academic Performance**: By providing personalized recommendations, students can optimize their study habits and improve their academic performance.

**Scalable Solution**: The methodology used in this project can be scaled and adapted for different academic disciplines and institutions.

**Further Research**: The study opens up avenues for further research into incorporating additional data sources and refining the model’s architecture.

### 4.8 Conclusion

In conclusion, this chapter presented the results of the data analysis and model performance evaluation. The model demonstrated strong performance in recommending personalized study timetables, with potential for further improvement. Key limitations and implications of the study were discussed, and suggestions for future work were provided.

# CHAPTER FIVE

## CONCLUSION AND RECOMMENDATIONS

## 5.0 Introduction

This chapter provides a detailed discussion of results of the study on developing the GPA Goal Achiever app, a personalized study plan recommender system, in facilitating the attainment of targeted CGPA by students pursuing Computer Science at UENR. Such discussions include the interpretation of results, comparison with existing studies, implications for education. Limitations of the study, recommendations of improvements for the future, and work possibly conducted in the near future are also discussed.

## 5.1 Discussions

### 5.1.1 Interpretation of results

Results obtained from the application of the GPA Goal Achiever app showed that the system was able to provide personalized study plans for students with respect to their current and target CGPA and study habits. Application of the machine learning algorithms, especially collaborative filtering, provided recommendations effectively to improve the academic performance. Data analysis also indicated that students who had clearly stated academic targets tended to benefit from the system by managing their time and improving their academic outcomes. Convergence and accuracy of the model, as reflected in MAE metrics, suggested that the app could provide enormous assistance in helping students reach their academic goals.

### 5.1.2 Comparison with existing studies

Results from the study are, therefore, a companion to other studies involving recommender systems in educational technology, such as those by (Ricci et al. 2015) and (Pane et al. 2015). Similar to previous studies, this research illustrates how critical personalized learning experiences are to improving student outcomes. However, different from most of the existing systems, which basically provide recommendations of learning resources, this study produced a personalized study plan considering workload and academic goals for each student to bridge the gap that was missing in previous research done by (Su et al. 2017). Meanwhile, with the inclusion of reinforcement learning techniques, this study extends further possibilities to recommender systems in education in light of the works presented by (Nguyen 2020).

### 5.1.3 Implications for education

Results obtained from the study will add much value to higher education. The personalized study plans, created with the help of the GPA Goal Achiever app by the students themselves, can result in improved academic performances by optimizing their study schedule and workloads. With the app, it would provide real-time feedback and adjustment to their individual needs-self-regulated learning that is very important to their success in demanding fields, such as computer science. It also makes available a scalable solution that may be adapted to different academic contexts, with a view to enabling institutions to improve student engagement and learning outcomes.

## 5.2 Limitations

Notwithstanding these good results, several limitations have been identified. First, the size of the dataset used for training the model was relatively small, which may consequently affect the generalization ability of the results. No social, psychological, or economic factors were considered in the study as influencing students' academic performances. Another notable challenge was the so-called "cold-start" problem in collaborative filtering, where limited data on new users results in poor accuracy of recommendations.

## 5.3 Recommendations

For future development, it is recommended that:

* It is recommended that larger and more diversified data be employed in the further development to enhance the robustness of the model.
* This means that a hybrid recommender system, which will actually combine the content-based filtering with collaborative filtering, has to be developed in order to overcome what was called the "cold-start" problem.
* This recommendation model has to go a step further to consider external factors that affect the lifestyle of a student, including psychological and social wellbeing.

This means that every higher education institution should replicate such personalized study planning tools that promote student success, especially in those areas where academic demands are excessively high.

## 5.4 Future Work

Future research will be needed to broaden the scope of this model into other academic disciplines.

This was followed by the exploration of more complex algorithms' integration that includes deep learning and hybrid models to achieve better recommendation accuracy.

It will explore how the addition of psychological and socio-economic variables might add value to the personalized learning process.

This is to test the functionality of the App GPA Goal Achiever for scalability and adaptability across various institutions.

## 5.5 Summary and conclusion

This present study has, thus, successfully developed and evaluated an individualized study plan recommender system for computer science students at UENR. The GPA Goal Achiever app proved capable of supporting students in achieving their academic goals through individual and adaptive study plans. This study befits the increasing literature pertaining to educational recommender systems and forms a bedrock for some improvements which the future promises for personalized learning technologies.

# REFERENCES

Aggarwal, C. C. (2016). Recommender systems: The textbook. Springer. <https://doi.org/10.1007/978-3-319-29659-3>

Dabbagh, N., & Kitsantas, A. (2012). Personal learning environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning. The Internet and Higher Education, 15(1), 3-8. <https://doi.org/10.1016/j.iheduc.2011.06.002>

Dembo, M. H., & Seli, H. (2016). Motivation and learning strategies for college success: A focus on self-regulated learning. Routledge.

Karimi, M., Javidan, R., & Bahrami, H. (2020). A personalized learning path recommender system based on a learner's knowledge level. International Journal of Information and Education Technology, 10(1), 30-35. <https://doi.org/10.18178/ijiet.2020.10.1.1337>

Pane, J. F., Steiner, E. D., Baird, M. D., & Hamilton, L. S. (2015). Continued progress: Promising evidence on personalized learning. RAND Corporation. <https://doi.org/10.7249/RR1365>

Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender systems: Introduction and challenges. In F. Ricci, L. Rokach, & B. Shapira (Eds.), Recommender systems handbook (pp. 1-34). Springer. <https://doi.org/10.1007/978-1-4899-7637-6_1>

Su, X., Duan, J., Zhang, Y., & Xu, G. (2017). An intelligent personalized recommendation system for online learning based on deep learning. Educational Technology & Society, 20(1), 300-311.

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734-749. <https://doi.org/10.1109/TKDE.2005.99>

Aggarwal, C. C. (2016). Recommender systems: The textbook. Springer. <https://doi.org/10.1007/978-3-319-29659-3>

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. Knowledge-Based Systems, 46, 109-132. <https://doi.org/10.1016/j.knosys.2013.03.012>

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4), 331-370. <https://doi.org/10.1023/A:1021240730564>

García, P., Amandi, A., Schiaffino, S., & Campo, M. (2009). Evaluating Bayesian networks' precision for detecting students' learning styles. Computers & Education, 53(2), 444-452. <https://doi.org/10.1016/j.compedu.2009.03.018>

Kardan, A. A., Sadeghi, H., Ghidary, S. S., & Aghabeigi, M. (2013). Prediction of student course selection in online higher education institutes using neural network. Computers & Education, 65, 1-11. <https://doi.org/10.1016/j.compedu.2013.01.015>

Khan, S. (2020). Enhancing academic performance: The role of study skills. Journal of Education and Practice, 11(13), 102-107.

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7(1), 76-80. <https://doi.org/10.1109/MIC.2003.1167344>

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In F. Ricci, L. Rokach, & B. Shapira (Eds.), Recommender systems handbook (pp. 73-105). Springer. <https://doi.org/10.1007/978-0-387-85820-3_3>

Manouselis, N., Drachsler, H., Verbert, K., & Santos, O. C. (2011). Recommender systems for learning. Springer. <https://doi.org/10.1007/978-1-4419-0536-3>

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533. <https://doi.org/10.1038/nature14236>

Muñoz-Merino, P. J., Fernández, L. M., Delgado Kloos, C., & Muñoz-Organero, M. (2015). A software architecture for the measurement of the effectiveness of student usage of learning resources. Journal of Educational Technology & Society, 18(3), 230-245.

Nguyen, T. T. (2020). Reinforcement learning in educational settings: Challenges and promises. Journal of Artificial Intelligence Research, 68, 371-405. <https://doi.org/10.1613/jair.1.11723>

Pane, J. F., Steiner, E. D., Baird, M. D., & Hamilton, L. S. (2015). Continued progress: Promising evidence on personalized learning. RAND Corporation. <https://doi.org/10.7249/RR1365>

Pu, P., Chen, L., & Hu, R. (2012). Evaluating recommender systems from the user's perspective: Survey of the state of the art. User Modeling and User-Adapted Interaction, 22(4-5), 317-355. <https://doi.org/10.1007/s11257-011-9115-7>

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), The adaptive web (pp. 291-324). Springer. <https://doi.org/10.1007/978-3-540-72079-9_9>

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.

Zheng, L., Zhang, X., & Li, Y. (2021). Personalized learning path recommendation in online learning systems. Computers in Human Behavior, 115, 106627. <https://doi.org/10.1016/j.chb.2020.106627>

Aggarwal, C. C. (2016). Recommender systems: The textbook. Springer. <https://doi.org/10.1007/978-3-319-29659-3>

Dabbagh, N., & Kitsantas, A. (2012). Personal learning environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning. The Internet and Higher Education, 15(1), 3-8. <https://doi.org/10.1016/j.iheduc.2011.06.002>

Nguyen, T. T. (2020). Reinforcement learning in educational settings: Challenges and promises. Journal of Artificial Intelligence Research, 68, 371-405. <https://doi.org/10.1613/jair.1.11723>

Pane, J. F., Steiner, E. D., Baird, M. D., & Hamilton, L. S. (2015). Continued progress: Promising evidence on personalized learning. RAND Corporation. <https://doi.org/10.7249/RR1365>

Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender systems: Introduction and challenges. In F. Ricci, L. Rokach, & B. Shapira (Eds.), Recommender systems handbook (pp. 1-34). Springer. <https://doi.org/10.1007/978-1-4899-7637-6_1>

Su, X., Duan, J., Zhang, Y. and Xu, G., 2017. An intelligent personalized recommendation system for online learning based on deep learning. Educational Technology & Society, 20 (1), pp. 300-311.

Berkovsky, S., Heckmann, D., & Kruppa, M. (2012). "Adaptive and Personalized Learning in Educational Environments." *Educational Technology & Society*, 15(2), 205–219.

Dewey, R., Rosé, C. P., & Ai, H. (2016). "Intelligent Tutoring and Continuous Adaptation for Personalized Learning." *IEEE Transactions on Learning Technologies*, 9(2), 182–195.

Drachsler, H., & Kalz, M. (2016). "The MOOC and Learning Analytics Innovation Cycle (MOLAC)." *Journal of Computer Assisted Learning*, 32(1), 16–21.

Drachsler, H., Hummel, H. G. K., & Koper, R. (2015). "Personal recommender systems for learners in lifelong learning networks." *Educational Technology & Society*, 8(4), 25–36.

García-Peñalvo, F. J., Conde, M. Á., & Alier, M. (2017). "Personalized Learning Ecosystems and Education." *International Journal of Emerging Technologies in Learning*, 12(2), 35–45.

Khribi, M. K., Jemni, M., & Nasraoui, O. (2009). "Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval." *Educational Technology & Society*, 12(4), 30–42.

Méndez, G. R., Ochoa, X., & Chiluiza, K. (2014). "Data-Driven Educational Recommendations for a Personalized Learning Environment." *IEEE Transactions on Learning Technologies*, 7(3), 165–174.

Nkambou, R., Mizoguchi, R., & Bourdeau, J. (2010). *Advances in Intelligent Tutoring Systems.* Springer.

Papanikolaou, K. A., & Grigoriadou, M. (2019). "Adapting to Learning Styles in Computer-based Educational Systems." *Computers & Education*, 134, 50–62.

Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (2015). *Recommender Systems Handbook.* Springer.

Romero, C., & Ventura, S. (2010). *Educational Data Mining: A Survey from 1995 to 2005.* Springer.

Sampson, D. G., Karagiannidis, C., & Kinshuk. (2014). "Personalized learning: Educational, technological and standardization perspective." *Computers in Human Behavior*, 32, 171–173.

Sun, J., & Zhang, Y. (2019). "A Collaborative Filtering Recommender System for Personalized Learning Resources." *Journal of Educational Technology Development and Exchange*, 12(1), 15–26.

Tang, T., & McCalla, G. I. (2005). "Smart Recommendation for an Evolving e-Learning System: Architecture and Experiment." *International Journal on E-Learning*, 4(1), 105–129.

Vasquez, A., & Chiang, H. (2016). "The Effectiveness of Personalized Learning in Higher Education." *Journal of Learning Analytics*, 3(2), 1–22.

# Appendix

1. **Model codes**

import pandas as pd

from sklearn.metrics.pairwise import cosine\_similarity

# Load timetable data

df = pd.read\_csv('timetable.csv')

# Calculate CGPA Improvement

df['CGPA Improvement'] = df['Target CGPA'] - df['Current CGPA']

# Create a pivot table: Student ID vs Subjects

pivot\_df = df.pivot\_table(index='Student ID', columns='Subjects', values='CGPA Improvement', fill\_value=0)

# Compute cosine similarity between students

similarity\_matrix = cosine\_similarity(pivot\_df)

# Convert similarity to DataFrame

similarity\_df = pd.DataFrame(similarity\_matrix, index=df['Student ID'], columns=df['Student ID'])

# Timetable recommendation function

def recommend\_timetable(student\_id, df, similarity\_df, top\_n=3):

similar\_students = similarity\_df[student\_id].sort\_values(ascending=False).index[1:top\_n+1]

recommended\_timetable = df[df['Student ID'].isin(similar\_students)][['Subjects', 'Days of the Week', 'Time Slots']]

return recommended\_timetable

# User input script

def get\_user\_input():

print("Enter your details:")

current\_cgpa = float(input("Current CGPA: "))

target\_cgpa = float(input("Target CGPA: "))

study\_habits = input("Study Habits (e.g., Reading, Writing, Visual): ")

level = int(input("Level (e.g., 100, 200, etc.): "))

semester = input("Semester (First/Second): ")

return current\_cgpa, target\_cgpa, study\_habits, level, semester

# Get user input

current\_cgpa, target\_cgpa, study\_habits, level, semester = get\_user\_input()

# Calculate a combined score for similarity

def calculate\_combined\_similarity(row, current\_cgpa, study\_habits, level, semester):

cgpa\_difference = abs(row['Current CGPA'] - current\_cgpa)

study\_habit\_score = 1 if row['Study Habits'] == study\_habits else 0

level\_score = 1 if row['Level'] == level else 0

semester\_score = 1 if row['Semester'] == semester else 0

return cgpa\_difference - (study\_habit\_score + level\_score + semester\_score)

# Calculate similarity score for all students

df['Combined Similarity Score'] = df.apply(calculate\_combined\_similarity, axis=1,

current\_cgpa=current\_cgpa,

study\_habits=study\_habits,

level=level,

semester=semester)

# Find the closest matching student based on combined similarity

closest\_student = df.sort\_values(by='Combined Similarity Score').iloc[0]

# Recommend timetable based on the closest student

student\_id = closest\_student['Student ID']

recommendation = recommend\_timetable(student\_id, df, similarity\_df, top\_n=3)

print("\nRecommended Timetable for You:")

print(recommendation)

1. **Sample a Recommended Timetable**

