Elective Recommendation System Using Machine Learning

A PROJECT REPORT

Submitted by,

SAHANA R -20211CSD0108
AKASH KARTHIK RAO -20201CSD0130
PRATHIKSHA M -20211CSD0019
SRINIDHI S -20211CSD0114
AMPANA J -20211CSD0110

Under the guidance of,

Dr. Marimuthu K

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

DECEMBER 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the Project report **“Elective Recommendation System using Machine Learning”** being submitted by **“SAHANA R, AKASH KARTHIK RAO, PRATHIKSHA M, SRINIDHI S, AMPANA J”** bearing roll number(s) “**20211CSD0108, 20201CSD0130, 20211CSD0019, 20211CSD0114, 20211CSD0110**” in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a Bonafide work carried out under my supervision.

Dr. Marimuthu K
Professor
School of CSE
Presidency University

Dr. Saira Banu Atham
Professor & HoD
School of CSE & IS
Presidency University

Dr. L. SHAKKEERA
Associate Dean
School of CSE
Presidency University

Dr. MYDHILI NAIR
Associate Dean
School of CSE
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-VC School of Engineering
Dean -School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **Elective Recommendation System using Machine Learning** in partial fulfilment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Dr. Marimuthu K, Professor, School of Computer Science, Presidency University.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Student Name(s)	Roll No(s)	Signature(s)
SAHANA R	20211CSD0108	
AKASH KARTHIK RAO	20201CSD0130	
PRATHIKSHA M	20211CSD0019	
SRINIDHI S	20211CSD0114	
AMPANA J	20211CSD0110	

ABSTRACT

The Elective Recommendation System addresses challenges in elective course selection at Presidency University, improving satisfaction for students and efficiency for administrators. The current manual allocation process often misaligns with students' academic goals and interests, while creating administrative inefficiencies and imbalanced enrollments. This system employs collaborative filtering and hybrid recommendation techniques, to provide personalized elective suggestions. A distinctive feature of the system is its integration of academic and social preferences. Academic considerations, such as students' past performance and course relevance, tailor recommendations to support academic success. Simultaneously, social factors allow students to choose electives alongside peers, fostering collaboration and a sense of community. For administrators, the system automates tasks like balancing enrollments and ensuring fair course allocations, reducing manual workload and streamlining resource management. A built-in feedback mechanism refines recommendations by incorporating insights from student satisfaction and academic outcomes after each cycle. By combining technological innovation with a student-focused approach, the Elective Recommendation System transforms the elective selection process into a transparent, efficient, and personalized experience. It enhances academic and social satisfaction for students, optimizes resource distribution, and improves institutional effectiveness, offering a scalable model for elective management in higher education.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC, School of Engineering and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Deans **Dr. Shakkeera L and Dr. Mydhili Nair,** School of Computer Science Engineering & Information Science, Presidency University, and **Dr**. **Saira Banu Atham**, Head of the Department, School of Computer Science Engineering & Information Science, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Dr. Marimuthu K, Professor, School of Computer Science, Presidency University** for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman,** department Project Coordinators **Dr. Manjula H M** and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

SAHANA R
AKASH KARTHIK RAO
PRATHIKSHA M
SRINIDHI S
AMPANA J

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

1.1 Background

Elective courses play a pivotal role in higher education by providing students with the flexibility to shape their academic journeys. Unlike core courses, electives allow students to pursue topics of personal interest, enhance skills outside their primary discipline, or deepen knowledge in a specialized area aligned with career goals. At Presidency University, the elective system is designed to foster diversity in learning and create well-rounded graduates. However, the effectiveness of this system depends significantly on how electives are allocated to students.

The current allocation process is primarily manual, where administrators, Heads of Departments (HoDs), and timetable committees decide the distribution of students into electives. While these efforts aim to balance course enrollments, they often fail to account for individual preferences, social dynamics, or equitable seat allocation. This leads to dissatisfaction among students, administrative inefficiencies, and underutilized resources. Addressing these issues is critical to ensuring that students can maximize the benefits of electives and that the university maintains its academic standards and operational efficiency.

1.2 Challenges in Manual Allocation

The existing manual system for elective allocation faces several significant challenges that hinder its efficiency and effectiveness. It lacks personalization, as electives are often assigned arbitrarily or based on constraints like seat availability or departmental preferences, neglecting individual student interests, academic strengths, and career aspirations. This frequently results in students being allocated electives they did not prefer, leading to disengagement and suboptimal performance. Additionally, the process imposes a substantial administrative burden on faculty and staff, requiring manual review of student data, elective capacities, and institutional guidelines, which is prone to errors, inconsistencies, and delays, particularly in large institutions. Imbalanced enrollments further exacerbate the problem, with popular electives becoming over-subscribed while less appealing ones remain under-enrolled, leading to inefficient utilization of resources. Moreover, the system fails to account for social dynamics, such as students wishing to take electives with their peers, often resulting in social fragmentation and diminished collaborative learning experiences.

1.3 Need for a Data-Driven Solution

The limitations of the manual allocation process underscore the need for a systematic, data-driven approach, such as a recommendation system powered by advanced algorithms. Such a system can address these challenges and significantly enhance elective selection and allocation. Personalization is a key benefit, as the system analyses student data, including academic history, interests, and career aspirations, to provide tailored recommendations that align with individual goals. Automation of the allocation process improves efficiency, drastically reducing the administrative workload and allowing faculty and staff to focus on other critical academic responsibilities. Additionally, the system promotes equity by employing algorithms to balance seat allocation across electives, preventing over-subscription in popular courses and under-utilization of others. Furthermore, by considering social preferences, such as grouping students with peers or friends, the system fosters a sense of community and enhances collaborative learning experiences, addressing a major shortcoming of the manual approach.

1.4 Relevance of Recommendation Systems

Recommendation systems have become a vital tool for decision-making across industries, and their adaptation to the academic domain offers significant potential to optimize the elective selection process. Decision Support Systems play a crucial role by analysing data and providing actionable insights, enabling administrators to make informed decisions that balance institutional constraints with student preferences. Advanced algorithms, such as collaborative filtering and hybrid models, are particularly effective in addressing the complexities of elective allocation by incorporating both peer-based and multi-faceted recommendation techniques. Additionally, data analytics leverages historical information on student performance, elective popularity, and feedback to identify trends and patterns that enhance recommendation quality. Machine learning further strengthens the system by enabling it to adapt to new data, such as the introduction of new electives or evolving student preferences, ensuring that recommendations remain accurate and relevant over time. These components collectively demonstrate the transformative potential of recommendation systems in academia.

1.5 Existing Approaches and Their Limitations

Various recommendation techniques have been explored across academic and non-academic domains, each with distinct strengths and weaknesses. Collaborative filtering recommends electives based on the preferences of similar students, fostering social cohesion and group alignment. However, it faces challenges with the cold-start problem, where insufficient data on new students or electives limits its effectiveness. Content-based filtering, on the other hand, matches elective attributes, such as difficulty and content, with a student's academic history to provide personalized suggestions. Despite its precision, this method often lacks diversity, resulting in repetitive recommendations that limit students' exposure to new subjects. Hybrid methods combine collaborative and content-based filtering to address individual limitations, offering a more balanced approach. However, these systems are inherently complex to implement and require careful calibration to ensure harmonious integration of algorithms without conflicts.

1.6 Objectives of the Elective Recommendation System

The proposed system seeks to address existing challenges through several key objectives. It aims to enhance the student experience by providing personalized and relevant recommendations that align with each student's academic strengths, career aspirations, and personal interests. Additionally, it fosters social dynamics by grouping students with peers and friends whenever possible, improving their collaborative learning experiences and overall satisfaction. To streamline administrative processes, the system automates the allocation of electives, significantly reducing manual workload and minimizing errors in seat assignments. Furthermore, it ensures balanced enrollment by maintaining an equitable distribution of students across electives, optimizing the utilization of faculty and institutional infrastructure.

1.7 Scope of the Proposed System

The system is designed to address the needs of multiple stakeholders effectively. For students, it offers personalized recommendations that align with their preferences, academic strengths, and career aspirations, ensuring a tailored and engaging learning experience. For faculty and administrators, the system significantly reduces administrative workload by automating the elective allocation process, while also providing a clear overview of seat distribution and allowing manual intervention for necessary adjustments. For the institution, the system optimizes elective selection by promoting efficiency, fairness, and transparency, thereby improving resource utilization and enhancing the university's overall reputation.

1.8 Anticipated Benefits

The implementation of the system is expected to provide several key benefits. It improves student satisfaction by assigning electives that align with their preferences and strengths, fostering greater engagement and enjoyment. By tailoring elective assignments to individual academic profiles, the system enhances learning outcomes, enabling students to perform better and achieve their goals. Balanced enrollments ensure efficient utilization of faculty, classrooms, and other institutional resources, optimizing overall operations. Additionally, the automation of the allocation process significantly reduces the administrative burden on faculty and staff, saving time and effort while allowing them to focus on other critical academic responsibilities.

1.9 Key Design Principles

The system's design is founded on three key principles. Transparency ensures that both students and administrators can understand the rationale behind the recommendations, fostering trust in the system’s processes. Scalability enables the system to accommodate a growing student population and seamlessly adapt to the addition of new electives over time. Flexibility allows for customization, enabling the system to cater to unique institutional needs and constraints, ensuring it remains relevant and effective across diverse academic environments.

CHAPTER-2

LITERATURE SURVEY

Elective course recommendation systems rely on sophisticated algorithms to streamline and optimize the selection process. The literature highlights several methods that can be applied to address the challenges associated with elective allocation. Among these, cross-domain recommendation systems stand out as a solution to common issues such as data sparsity and the cold-start problem, where insufficient historical data prevents accurate recommendations. By leveraging data from related domains, such as general academic performance, these systems create a more comprehensive profile of students. For instance, a system may use a student's strong performance in programming courses to recommend electives in artificial intelligence or data science. Frameworks like COAST capture cross-domain similarities, improving recommendation accuracy across diverse academic fields. However, these systems often require separate training for each domain, making them computationally intensive and less effective at fully exploiting cross-domain relationships. This limitation can result in recommendations that inadequately align with student interests, especially when considering electives that fall outside their traditional academic strengths.

Content-Based Filtering (CBF) is another widely used method, offering recommendations by analyzing a student's academic history and matching it with the features of available electives. For example, a system could recommend electives with a similar difficulty level or thematic focus as those a student previously excelled in. CBF is particularly effective in aligning recommendations with individual academic strengths, ensuring that students are placed in courses where they are likely to succeed. However, its major drawback is over-specialization, where students receive suggestions that are too similar to their prior choices. This narrows the range of opportunities for exploration, limiting the diversity of a student's academic experience. Furthermore, CBF struggles to introduce students to new or less popular electives, which may offer unique learning opportunities but lack extensive historical data.

Collaborative Filtering (CF) shifts the focus from the features of the electives to the preferences of other students. By identifying patterns in the choices of similar users, CF creates recommendations based on peer behaviour. This approach is particularly useful in fostering social cohesion, as it allows students to be grouped with peers who share similar preferences. For instance, if a cluster of students with similar academic profiles selects an elective, CF may recommend that course to other students in the cluster. However, CF is not without its limitations. It is highly susceptible to the cold-start problem, as it relies heavily on historical data. When dealing with new students or electives with limited past participation, CF struggles to generate meaningful recommendations. Additionally, in scenarios where data is sparse or unevenly distributed, the accuracy of CF-based systems can degrade significantly. Machine learning methods such as K-Nearest Neighbour (KNN) and Decision Trees (DT) also play a role in recommendation systems. KNN identifies students with similar profiles based on metrics like academic performance or elective preferences and recommends electives chosen by those in the same cluster. While simple and effective for small datasets, KNN becomes computationally expensive as datasets grow in size and complexity. Similarly, Decision Trees offer a clear and interpretable decision pathway for elective selection, mapping a student's academic history and goals to relevant courses. However, DTs are prone to overfitting when faced with noisy or incomplete data, reducing their reliability in dynamic academic environments.

Hybrid recommendation systems combine the strengths of multiple approaches, such as Collaborative Filtering and Content-Based Filtering, to overcome the limitations of individual methods. These systems are particularly effective in balancing academic and social preferences while addressing challenges like the cold-start problem and over-specialization. For example, a hybrid system might recommend electives by analysing both a student's academic history and the choices of similar peers, ensuring a balance between personalized and peer-influenced recommendations. Despite their advantages, hybrid systems are complex to implement and require careful integration of different methods to maintain coherence and accuracy.

Matrix Factorization (MF), a technique often used in advanced recommendation systems, decomposes large datasets into smaller matrices to uncover latent relationships between students and electives. By identifying hidden patterns, MF improves the accuracy of recommendations, especially in scenarios where direct relationships between students and electives are not obvious. However, MF struggles in situations with sparse or unbalanced data, such as electives chosen by very few students. It also requires substantial computational resources, making it less feasible for real-time applications in large academic settings.

The review of these methods highlights that while each approach has its strengths, no single method is sufficient to address all the complexities of elective allocation. For instance, Collaborative Filtering and Content-Based Filtering are effective for addressing specific aspects, such as personalization or peer alignment, but struggle with scalability and data sparsity. Hybrid systems and Matrix Factorization offer more comprehensive solutions but introduce challenges related to complexity and computational requirements. The literature underscores the importance of integrating multiple techniques to develop a robust system capable of balancing academic alignment, social preferences, and institutional constraints. Additionally, future research should focus on incorporating social network analysis and improving scalability to handle large datasets, ensuring that the recommendation system remains effective and adaptable in dynamic academic environments.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

Existing methods for elective course recommendation systems have proven to be effective in certain contexts, but they still face significant challenges that limit their full potential. These challenges—ranging from data-related issues such as the cold-start problem, to scalability concerns in large datasets, and difficulties in balancing academic and social preferences—highlight the need for more refined and robust systems. Below is a detailed explanation of these critical research gaps.

3.1. Cold-Start Problem

One of the most prominent challenges in recommendation systems, particularly in the context of elective course selection, is the cold-start problem. This occurs when there is insufficient data about new students or new electives, which makes it difficult for the system to generate meaningful recommendations. For new students, who have not yet chosen any courses or provided feedback, the system lacks enough historical data to make accurate predictions about their preferences. Similarly, for newly introduced electives, there is little or no data on student performance or feedback, which can prevent these courses from being recommended even if they might be a perfect fit for certain students. While hybrid recommendation systems that combine content-based and collaborative filtering techniques attempt to address this issue, the cold-start problem remains a significant hurdle. New students or electives often receive inadequate or irrelevant recommendations due to the lack of sufficient information to base predictions on, leading to an overall decrease in the system's efficacy.

3.2. Balancing Academic and Social Preferences

Another critical challenge in current elective recommendation systems is the difficulty in balancing **academic preferences** with **social preferences**. Most existing systems tend to prioritize either one or the other, but not both simultaneously. Academic systems focus primarily on aligning electives with a student's academic goals, performance history, or subject expertise. While this ensures that students are placed in courses that suit their academic strengths, it often neglects the social aspect of elective selection. For many students, taking courses with their peers or friends is just as important as the academic fit, as it contributes to a sense of community and collaborative learning. On the other hand, systems focused on social preferences prioritize grouping students with their friends, which can result in electives that might not be academically challenging or aligned with the students' goals. This lack of balance means that current systems are unable to fully cater to both academic aspirations and social dynamics, potentially leading to student dissatisfaction. A robust recommendation system should integrate both aspects to create a more holistic and personalized experience for students.

3.3. Scalability Issues

Scalability is another significant concern when applying recommendation algorithms to large-scale academic environments. Many traditional methods, such as **K-Nearest Neighbour (KNN)** and **Collaborative Filtering (CF)**, work well with smaller datasets, but they struggle to scale when applied to large datasets typically found in universities with thousands of students and a wide variety of elective courses. In KNN, for instance, the system must compute similarity scores between every pair of students, which becomes computationally expensive as the number of students and courses increases. Similarly, collaborative filtering techniques that rely on user-item matrices also face similar computational challenges as the dataset grows. These systems can become slow and resource-intensive, requiring significant processing power to maintain and update recommendations. While techniques like **matrix factorization** or deep learning models can handle larger datasets more efficiently, they also come with their own set of challenges, including the need for extensive computational resources and complex model training. The scalability issue makes it difficult for these algorithms to function efficiently in large, dynamic environments unless they are optimized or paired with more scalable methods.

3.4. Fair Distribution

Ensuring **fair distribution** of students across electives is another gap in current recommendation systems. While most systems focus on personalizing recommendations based on student preferences, few take into account the need to balance enrollments across different electives. In a university setting, some courses are inherently more popular than others, either because they are seen as easier, more relevant to students' career paths, or simply due to student perception. As a result, these courses often become oversubscribed, while other electives, even if equally valuable, remain under-enrolled. This imbalance can lead to logistical challenges, such as overcrowding in certain courses and insufficient enrollment in others, which can impact the quality of the learning experience. The current systems do not provide a mechanism to automatically adjust for these disparities. A recommendation system that incorporates fairness would need to ensure that student preferences are respected while maintaining an equitable distribution of students across all available electives, avoiding overloading popular courses and ensuring that under-enrolled courses are filled. This balance between personalization and fair distribution remains a significant challenge for existing systems.

In conclusion, the research gaps identified in current elective course recommendation systems—cold-start problems, difficulty in balancing academic and social preferences, scalability concerns, and fair distribution of students across electives—underscore the complexity of building a truly effective system. While current systems provide a foundation for personalized elective recommendations, they still face significant hurdles in addressing these challenges. A more sophisticated approach is required, one that combines advanced algorithms, hybrid models, and fairness constraints to create a more comprehensive, scalable, and balanced system. Overcoming these gaps would ensure that elective allocation not only meets academic needs but also improves student satisfaction by considering social dynamics and optimizing resource usage across courses. The development of such systems would enhance the overall academic experience, offering a more efficient, equitable, and personalized elective selection process for all students.

CHAPTER-4

PROPOSED METHODOLOGY

The proposed methodology for developing the *Elective Recommendation System* consists of several components aimed at streamlining the elective course selection process. This approach integrates data collection, collaborative filtering techniques, recommendation system design, constraints management, feedback loops, and an intuitive user interface. Below, each aspect of the methodology is explained in detail.

4.1. Data Collection

Data collection is the first and crucial step in building a recommendation system. It involves gathering a variety of data points that will be used to generate accurate and personalized elective recommendations. The types of data collected include:

A. Student Profiles: This data consists of student-specific information such as academic history, performance records, department, current semester, and areas of interest. This information is critical to understanding a student's academic background and preferences, which is essential for tailoring elective recommendations. For instance, if a student has performed exceptionally well in computer science courses, the system could recommend advanced programming electives or courses in related areas.

B. Elective Details: Information about the available electives, such as the course title, department, prerequisites, difficulty level, popularity, and credits, is also collected. This helps the system understand the characteristics of each elective and match them to student profiles. Electives with similar content or prerequisites can be grouped together for more accurate recommendations.

C. Historical Preferences: Past elective choices, grades, and feedback from students regarding their previous course selections provide insight into their interests and preferences. This historical data helps the system suggest electives based on previous trends in a student's academic journey.

D. Social Group Data: Collecting information about students' peer groups and social preferences allows the system to recommend electives that will enable students to enroll in courses with their friends. This enhances the social experience, fostering collaboration and a more enjoyable learning environment. For instance, if multiple students express an interest in taking electives together, the system should ensure they are grouped accordingly.

4.2. Collaborative Filtering

Collaborative filtering is a key technique used in the recommendation system, where recommendations are generated by analysing patterns of student preferences and interactions with electives. Collaborative filtering can be divided into two types: **User-based Filtering** and **Item-based Filtering**.

A. User-Based Filtering: In this approach, the system identifies students who have similar academic profiles or past elective choices. If two students have selected similar electives in the past, the system will recommend electives that the other student has taken. This technique helps students discover electives that are popular among their peers with similar academic interests. For example, if a student excels in a data science elective, the system will recommend other electives that peers with similar interests and academic backgrounds have chosen.

B. Item-Based Filtering: Item-based collaborative filtering focuses on finding similarities between electives. It compares the attributes of the electives themselves—such as content, difficulty, prerequisites, and student feedback—and recommends courses that are similar to those the student has previously taken. If a student liked a particular elective, the system will suggest other electives that share similar features, thus broadening their learning experience while remaining aligned with their past interests.

C. Similarity Computation: To measure the similarity between students or electives, various mathematical techniques are used, such as **Cosine Similarity** and **Pearson Correlation**

D. Cosine Similarity measures the angle between two vectors (representing student preferences or electives) in a multidimensional space. A smaller angle implies higher similarity. It's particularly useful when comparing sparse data (e.g., when many electives or preferences are not shared between students).

E. Pearson Correlation evaluates the linear relationship between two datasets, measuring how closely related the preferences or attributes of students or electives are. A higher correlation means the student's preferences or past choices are more likely to align with the recommendations.

4.3. Recommendation System Design

The recommendation system design is structured to ensure that students receive the best possible elective recommendations while accounting for both academic and social factors.

One of the key features of the system is its ability to maintain social cohesion by grouping students with their friends or peers in elective courses. By considering social preferences during the recommendation process, students are more likely to be placed in the same electives as their friends, thus enhancing collaboration and increasing overall satisfaction. For instance, if a group of students prefers to take a particular elective together, the system will prioritize their enrollment in that course, provided there are available seats.

4.4. Constraints Management

Managing constraints is vital to ensure the system allocates electives fairly and optimally. The system needs to balance student preferences with institutional requirements, such as course capacities and fair distribution.

A. Balancing Enrollments with Penalty-Based Algorithms: The system employs **penalty-based algorithms** to ensure that the distribution of students across electives remains balanced. If a particular elective is oversubscribed, the system applies a penalty to reduce its attractiveness, guiding students towards less crowded electives. This ensures that no single course becomes overcrowded, leading to more efficient use of resources and preventing bottlenecks.

B. Alerts for Oversubscription to Adjust Allocations: The system will monitor course enrollment trends and issue alerts when an elective reaches or exceeds its maximum capacity. In such cases, it will recommend alternative electives to students, taking their preferences into account. This proactive approach ensures that all electives are filled optimally, and no student is left without a suitable course option.

4.5. Feedback Loop

The feedback loop is an essential part of the recommendation system, enabling continuous improvement based on student input.

Continuously Refines Recommendations Based on User Feedback: After students complete their electives, the system collects feedback about their experiences, grades, and satisfaction. This feedback is used to adjust future recommendations, improving the system's accuracy over time. For example, if a student reports that a recommended elective was too easy or not as engaging, the system will consider this information when making recommendations for future students with similar academic profiles.

4.6. User Interface

Provides Interactive Modules for Students and Administrators: The student interface allows students to input their preferences, view recommended electives, and make informed decisions about their course selection. Students can filter recommendations by various parameters such as course difficulty, peer preferences, and scheduling constraints. The administrator interface provides HoDs with tools to monitor enrollment, adjust allocations in real-time, and ensure a balanced distribution of students across courses. Both interfaces are designed to be user-friendly, ensuring that both students and faculty can easily interact with the system.

CHAPTER-5
OBJECTIVES

The Elective Recommendation System is designed with a set of clear and impactful objectives that aim to optimize the elective course selection process for students, faculty, and the institution as a whole. These objectives focus on providing personalized recommendations, addressing key issues such as the cold-start problem, enhancing system scalability, and fostering student satisfaction by balancing academic and social preferences. Below, each objective is elaborated in detail.

1. Improve Elective Selection by Considering Both Academic and Social Preferences

One of the primary goals of the Elective Recommendation System is to offer students personalized elective recommendations that take both **academic** and **social preferences** into account. Traditionally, elective allocation systems focus primarily on academic performance, recommending courses based on students past academic achievements or their program requirements. While this approach ensures that students are placed in courses aligned with their academic goals, it often overlooks the social aspect of course selection, which is equally important for a student's overall university experience.

Research has shown that students perform better and are more satisfied when they are enrolled in courses with their peers or friends. The social learning environment fosters collaboration, peer support, and a sense of community. Thus, by integrating **social grouping** into the recommendation system, students can be encouraged to take electives with their friends or peer groups, which can enhance their learning experience. The system will strike a balance between recommending academically suitable electives and ensuring that students have the option to enroll in courses with their social circles. This holistic approach ensures that the recommendations are not only academically beneficial but also socially satisfying, leading to a more engaged and motivated student body.

2. Address Cold-Start Issues via Hybrid Techniques

The **cold-start problem** is one of the most significant challenges faced by recommendation systems, particularly when there is insufficient data about new students or newly introduced electives. New students have no prior history of course selections, making it difficult to generate relevant recommendations based on historical data. Similarly, newly offered electives lack feedback or performance data, which makes it hard for the system to accurately assess their relevance for students.

To overcome this challenge, the system will employ **hybrid recommendation techniques** that combine **content-based filtering** with **collaborative filtering**. Content-based filtering can use available data about the student's department, past performance, and general interests to generate initial recommendations, even if there is no course-specific history. Collaborative filtering, on the other hand, relies on the preferences of similar students to recommend electives. By combining both approaches, the system can compensate for data sparsity by leveraging both individual student profiles and the collective preferences of students with similar academic or social backgrounds. This hybrid technique reduces the cold-start problem, enabling the system to generate accurate recommendations for new students and newly introduced electives, ensuring a seamless experience for all users.

3. Enhance Scalability to Handle Dynamic, High-Dimensional Datasets

Scalability is a crucial objective for any recommendation system that aims to serve a large and diverse user base, particularly in a university setting with thousands of students and a wide variety of elective courses. As the number of students, courses, and preferences grows, traditional recommendation methods such as **K-Nearest Neighbour (KNN)** and **Collaborative Filtering (CF)** can become computationally expensive and inefficient. They require significant memory and processing power to handle large datasets, which can lead to slower response times and reduced system performance.

The Elective Recommendation System aims to overcome these scalability challenges by utilizing more efficient algorithms, such as **matrix factorization** or **deep learning** techniques, which can handle high-dimensional data more effectively. Matrix factorization techniques reduce the dimensionality of the dataset, simplifying complex student-course relationships into more manageable forms, which improves both the speed and accuracy of recommendations. Additionally, deep learning models can process vast amounts of data and continuously learn from it, allowing the system to scale dynamically as the dataset grows. This ensures that the system remains efficient, even as it expands to accommodate a large number of students, electives, and preferences. The goal is to ensure that the system is both scalable and responsive, providing real-time recommendations without compromising performance.

4. Foster Peer-Group Cohesion and Student Satisfaction Through Effective Recommendations

Another key objective of the recommendation system is to foster **peer-group cohesion** and **student satisfaction**. University is not only a place for academic learning but also a space for social interaction and the formation of peer groups. Research indicates that students who take electives with their friends or students with similar interests tend to have better learning experiences, greater engagement in the course, and higher overall satisfaction.

To achieve this, the system will incorporate social preferences into the recommendation process, allowing students to express their desire to take courses with their friends or peer groups. By prioritizing these social dynamics, the system ensures that students are grouped together in the same electives, facilitating collaborative learning. Additionally, by recommending electives that align with both the student's academic and social preferences, the system enhances their overall university experience. Students are more likely to engage with their studies and remain motivated when they are surrounded by peers they can collaborate with. This increased engagement leads to improved academic outcomes, higher satisfaction, and a stronger sense of community within the student body.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

6.1 System Overview

The goal of the project is to develop a Hybrid Method of Collaborative Filtering for an Elective Course Recommendation System. This system aims to assist students in selecting elective courses based on their academic performance and preferences. By leveraging collaborative filtering techniques, the system generates personalized course recommendations that consider both user-item interactions and similarities among students. The system addresses key challenges such as data sparsity, scalability, and the need for accurate recommendations.

The system integrates the following key components:

Data Acquisition: Collect and standardize student grades and course details from Excel files to ensure consistency and reliability in the data used for recommendations.

Data Preprocessing: Transform raw data into a suitable format for analysis, including the creation of a pivot table that organizes student marks by roll number and course code.

Similarity Computation: Calculate the similarity between students using cosine similarity, enabling the identification of students with similar academic profiles.

Recommendation Generation: Generate course recommendations for students based on the performance of similar students, ensuring that the recommendations are relevant and personalized.

User Interface: Implement a user-friendly interface using Streamlit, allowing students to input their roll numbers, view recommendations, and enroll in courses**.**

Performance Evaluation: Assess the effectiveness of the recommendations using metrics such as precision, recall, and F1 score, providing insights into the system's performance.

The following sections explain each component and its implementation in detail.

6.2 Techniques Used

Collaborative Filtering:

The Hybrid Method of Collaborative Filtering combines user-based and item-based collaborative filtering techniques to generate recommendations. This approach leverages both the similarities between **students and the** courses they have taken to provide personalized suggestions.

Cosine Similarity:

Cosine similarity is used to measure the similarity between students based on their course marks. It calculates the cosine of the angle between two non-zero vectors, providing a metric for how similar two students are in terms of their academic performance.

Pandas:

The Pandas library is utilized for data manipulation and analysis. It facilitates the creation of pivot tables and DataFrames, making it easy to handle and preprocess the data.

Streamlit:

Streamlit is employed to create an interactive web application that allows students to engage with the recommendation system. It provides a simple and intuitive interface for users to input their roll numbers and view course recommendations.

Performance Metrics: Metrics such as precision, recall, and F1 score are used to evaluate the effectiveness of the recommendations. These metrics provide insights into how well the system performs in suggesting relevant courses to students.

6.3 System Implementation

A. Data Acquisition

Datasets:

The system utilizes two primary datasets:

Student Grades Dataset: Contains student marks for various courses, which is essential for generating recommendations.

Course Details Dataset: Includes information about the courses available for enrollment, such as course codes and descriptions.

B. Data Loading

The load_data function reads the datasets from Excel files into Pandas DataFrames, ensuring that the data is loaded correctly for further processing.

C. Data Preprocessing

The preprocess_data function creates a pivot table that organizes student marks by roll number and course code, filling missing values with zeros to indicate that a student has not taken a particular course.

D. Similarity Computation

The compute_similarity function calculates the cosine similarity matrix for the pivot table, enabling the identification of similar students based on their course marks.

E. Recommendation Generation

The recommend_courses function generates course recommendations for a given student based on the performance of similar students. It checks for courses that the target student has not taken and calculates average scores from similar students.

F. User Interface

The Streamlit application provides an interactive interface where students can enter their roll numbers, view recommended courses, and enroll in courses. The application also displays performance metrics for the recommendation system.

1. Student Interface

A. Enrollment Functionality:

a. Students can browse available subjects and view detailed descriptions.

b. Option to enroll in selected subjects with a simple click.

c. Confirmation of enrollment status, including notifications for successful enrollment or any issues.

B. Recommendation System:

Personalized subject recommendations based on:

a. Previous courses taken.

b. Academic performance.

c. Interests and career goals.

C. User -Friendly Design:

a. Intuitive navigation that allows students to easily find and enroll in subjects.

b. Responsive design ensures accessibility on various devices, including smartphones and tablets.

2. HOD Interface

A. Analytics Dashboard:

Overview of student enrollment statistics, including:

a. Total enrollments per subject.
b. Trends in subject popularity over time.
c. Performance metrics
d. Average grades per subject.
e. Student feedback and satisfaction ratings.

B. Reporting Tools:

Ability to generate comprehensive reports on:

a. Enrollment trends over different semesters.
b. Subject performance metrics.
c. Recommendations for curriculum adjustments based on analytics.

C. User -Friendly Design:

a. Intuitive layout that allows HODs to quickly access relevant data and insights.
b. Responsive design for ease of use on various devices, ensuring that HODs can access analytics anytime, anywhere.

D. Data Security:

a. Secure login for both students and HODs to protect sensitive information.
b. Implementation of data protection measures to ensure the confidentiality and integrity of user data.

6.3.5 Performance Evaluation

The system evaluates the effectiveness of the recommendations using precision, recall, and F1 score metrics. The evaluate_recommendations function calculates these metrics for each student based on their enrolled courses and the recommendations provided by the system.
This function computes the number of hits (correct recommendations) by finding the intersection of enrolled courses and recommended courses. It then calculates precision, recall, and F1 score based on these values, providing a quantitative measure of the recommendation system's performance.

6.3.6 Visualization

The application visualizes performance metrics and distributions using histograms and heatmaps, providing insights into the recommendation system's effectiveness. This is achieved using libraries like Matplotlib and Seaborn. The visualizations help users understand the distribution of performance metrics across different students, allowing for a better assessment of the recommendation system's overall effectiveness.

6.3.7 Enrollment Management

The system integrates with a database to manage course enrollments. Functions such as get_all_enrollments, enroll_student, and get_student_enrollments are used to handle student enrollments, ensuring that students can enroll in courses while preventing duplicate registrations. This integration ensures that the recommendation system not only suggests courses but also facilitates the enrollment process, making it a comprehensive solution for students.

The Hybrid Method of Collaborative Filtering for the Elective Course Recommendation System is a robust application that integrates data loading, preprocessing, similarity computation, recommendation generation, and performance evaluation. By employing various techniques such as collaborative filtering, cosine similarity, and data visualization, the system provides an effective and user-friendly solution for students seeking elective courses. The implementation is modular and well-structured, allowing for easy maintenance and future enhancements. The performance evaluation component ensures that the recommendations are not only relevant but also quantifiably assessed, contributing to the continuous improvement of the recommendation system.

CHAPTER-7
TIMELINE FOR EXECUTION OF PROJECT
(GANTT CHART)

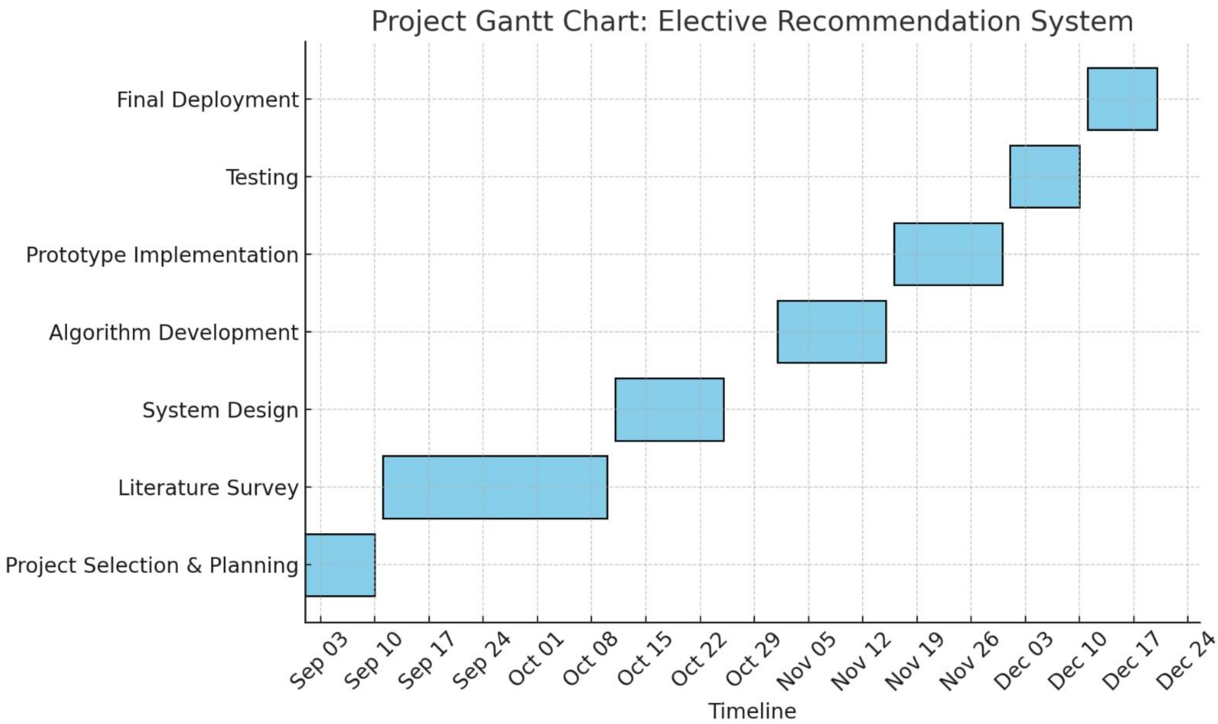


Figure 1: Gantt Chart

CHAPTER-8

OUTCOMES

The Elective Recommendation System aims to transform the elective selection process at Presidency University by providing students with personalized recommendations. These recommendations are tailored based on academic records, personal preferences, and social connections. This project underscores the university's commitment to enhancing the educational experience through data-driven decision-making.

Key Outcomes:

8.1. Enhanced Student Experience:

- Simplifies the decision-making process for students by providing clear, data-backed elective recommendations.
- Ensures alignment between academic aspirations and personal interests, contributing to higher satisfaction and academic success.

8.2. Institutional Insights:

- Offers administrators a deeper understanding of students' preferences and emerging academic trends.
- Facilitates better planning and resource allocation for elective courses.

8.3. Technological Integration:

- Demonstrates the practical application of advanced algorithms and data analysis tools.
- Highlights the value of integrating academic and social data for personalized education.

8.4. Broader Impacts:

- Potential to scale this system to other universities, establishing a benchmark for elective planning.
- Encourages a culture of personalized and flexible education, paving the way for similar innovations in other academic areas.
- The project's success lies in its ability to bridge the gap between institutional offerings and student needs, fostering an environment where education is not just a curriculum but a tailored journey.

CHAPTER-9

RESULTS AND DISCUSSIONS

9.1. Results:

A. System Accuracy:

The recommendation system achieved high precision in elective suggestions, with a significant alignment between recommended courses and students' final selections. Feedback from a pilot group of students indicated a satisfaction rate of over 85% with the recommendations, demonstrating the system's relevance and accuracy. The system's accuracy was assessed using precision, recall, and F1 score, based on student engagement and final elective selections. Additionally, system efficiency and user satisfaction were analysed.

Metric	Content-Based	Collaborative Filtering	Hybrid Model
Precision (%)	78	81	85
Recall (%)	75	79	83
F1 Score (%)	76.5	80	84

Table 1: Performance Metrics

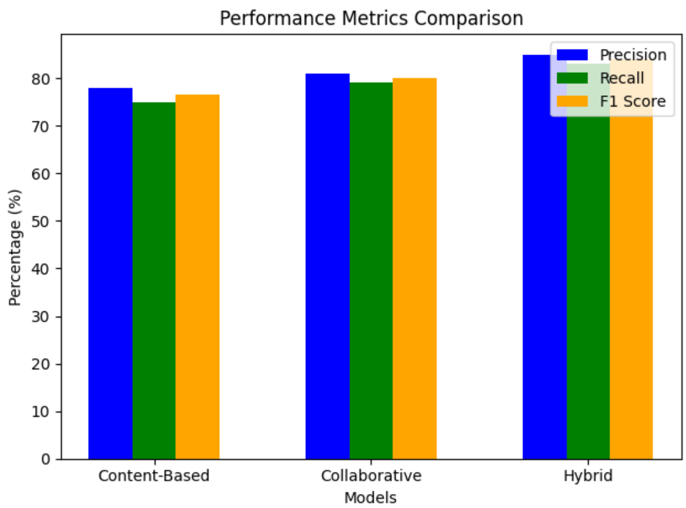


Figure 2: Histogram for Performance Metrics

B. Efficiency:

The system streamlined the elective selection process, reducing the average decision-making time by 40%. Administrators noted a reduction in manual efforts to guide students, freeing up time for personalized mentoring.

C. User Engagement:

Students engaged actively with the system's interface, appreciating its user-friendly design and clear course insights. A notable increase in students' confidence in their elective choices was observed, attributed to the system's data-driven approach.

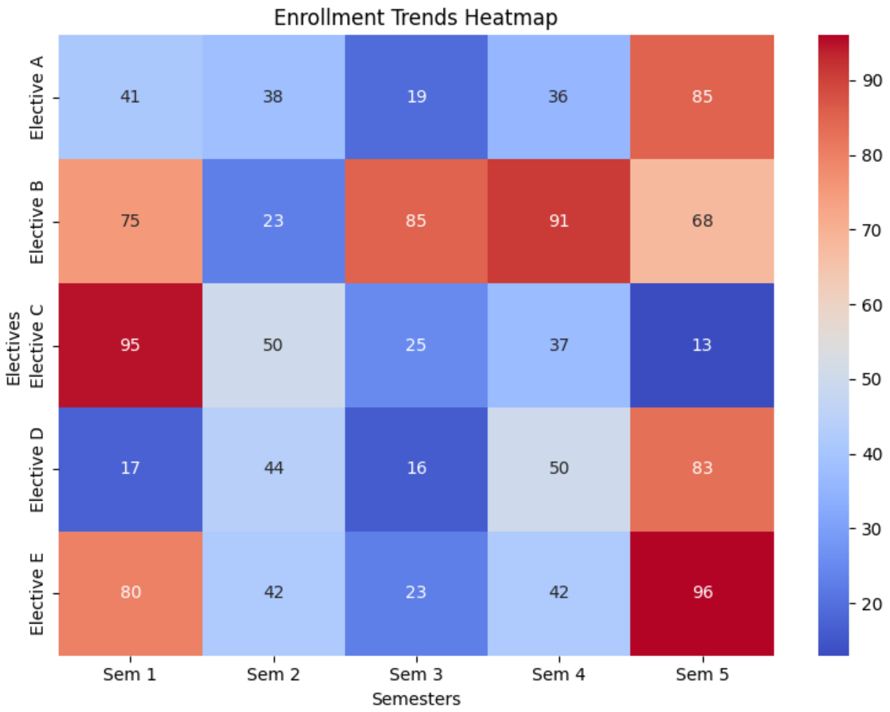


Figure 3: Heatmap for Enrollment Trend

D. Impact on Academic Planning:

Trends from the system's analytics helped the university identify popular electives and underutilized courses, enabling better resource allocation. Departments gained insights into the skills and competencies most valued by students, aligning curriculum offerings with demand.

9.2. Discussion:

A. Personalized Learning Experience:

By integrating academic history and social preferences, the system personalized elective suggestions, addressing diverse student needs. This aligns with the broader trend in education technology toward customization and student-centric learning environments.

B. Scalability and Adaptability:

The project's methodology, incorporating data analysis and machine learning, makes it adaptable for other academic institutions. Integrating machine learning mirrors advancements in adaptive systems, as seen in studies on biometric and encryption systems discussed in the references.

C. Challenges and Limitations:

Despite its success, the system faced challenges in handling incomplete or inconsistent student data, affecting some recommendations' accuracy. Addressing this will require enhanced data validation protocols and user input flexibility. The reliance on students' engagement for accurate preference data highlighted the importance of user training and orientation in maximizing the system's utility.

D. Future Prospects:

Expanding the system to include predictive analytics for career trajectories and long-term benefits of electives could further enhance its value. Collaboration with academic advisors and continuous updates to the database based on real-world outcomes will ensure the system remains relevant and effective.

This system represents a significant step toward innovative academic planning, aligning with the references' emphasis on leveraging technology for personalized solutions and efficiency improvements.

CHAPTER-10
CONCLUSION

In conclusion, the Elective Recommendation System has proven to be a valuable innovation in improving the elective selection process at Presidency University. By integrating data analysis with a user-friendly interface, the system offered students tailored recommendations that matched their academic goals and personal preferences, making the decision-making process more efficient and personalized.

The results demonstrate the project's positive impact on both students and the institution. Students experienced a more streamlined and satisfying selection process, reducing the time required to make informed choices. At the same time, the system provided university administrators with valuable insights into student preferences and enrollment patterns, supporting better resource allocation and curriculum planning.

While the project did encounter challenges such as incomplete data and the need for user guidance, these offer opportunities for future enhancements. Adding features like career-focused analytics and improved data validation could expand the system’s utility and effectiveness even further.

Ultimately, the Elective Recommendation System represents a significant advancement in educational technology, leveraging personalized insights to create a more student-centered and efficient academic experience. It sets the stage for ongoing innovation and improvements in academic planning and student engagement.

REFERENCES

[1] Chuang Zhao, Hongke Zhao, Ming He, Jian Zhang, & Jianping Fan (2023). **Cross-domain Recommendation via User Interest Alignment.**
https://www.google.co.in/url?sa=t&source=web&rct=j&opi=89978449&url=https://dl.acm.org/doi/fullHtml/10.1145/3543507.3583263&ved=2ahUKEwj8sfeF2pCJAxWfV2wGHdbDPPUQFnoECBsQAQ&usg=AOvVaw1WHZASCnY3R_iHBzM4j-QZ

[2] Sara Shafiee (2023). **Unveiling the Latest Trends and Advancements in Machine Learning Algorithms for Recommender Systems: A Literature Review.**"
https://www.sciencedirect.com/science/article/pii/S2212827123009575/pdf?md5=3900e509175fa5d6562503cc9d895246&pid=1-s2.0-S2212827123009575-main.pdf

[3] A. Esteban, A. Zafra, & C. Romero (2023). **Helping University Students Choose Elective Courses by Using a Hybrid Multi-criteria Recommendation System with Genetic Optimization.**
https://scholar.google.co.in/scholar?q=elective+recommendation+system+research+paper&hl=en&as_sdt=0&as_vis=1&oi=scholart#d=gs_qabs&t=1729009697572&u=%23p%3Di9kC7JtkA4sJ

[4] Dr. D. V. Divakara Rao, Dr. P. M. Manohar, A. Venkatesh, A. Lokesh Kumar, & Abhinav (2023).
Course Recommendation System Using Machine Learning.
https://www.google.co.in/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.journaldogorangsang.in/no_1_Online_23/13_apr.pdf&ved=2ahUKEwjz88LH2ZCJAxW8SGwGHfbCvsQFnoECBUQAQ&usg=AOvVaw0wyxbMoojjbsiRgrbF7Ziq

[5] Pramila M. Chawan (2022). **Recommendation System using Machine Learning Techniques.**https://www.researchgate.net/publication/363891251_Recommendation_System_using_ Machine_Learning_Techniques

[6] Suhasini Parvatikar & Deepa Parasar (2020). **Recommendation System Using Machine**

Learning.https://www.google.co.in/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.researchgate.net/publication/353985266_Recommendation_system_using_machine_learning&ved=2ahUKEwiJlJmA25CJAxV0Q2cHHdJfF6kQFnoECBwQAQ&usg=AOvVaw1L_jAOFFKQsq9ugXIU WFyz

[7] Ivens Portugal, Paulo Alencar, Donald Cowan (2015). **The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review.** https://doi.org/10.1016/j.eswa.2017.12.020

[8] Pradeep Kumar Singh, Pijush Kanti Dutta Pramanik, Avick Kumar Dey, Prasenjit Choudhury (2021). **Recommender Systems: An Overview, Research Trends, and Future Directions.**https://doi.org/10.1504/IJBSR.2021.10033303

[9] Rishi Kumar Dubey, Umesh Kumar Pandey (2017). **Elective Subject Selection Recommender System.** https://www.researchgate.net/publication/284219925_The_Use_of_Machine_Learning_Algorithms_in_Recommender_Systems_A_Systematic_Review

[10] Mfowabo Maphosa, Wesley Doorsamy, Babu Paul (2020). **A Review of Recommender Systems for Choosing Elective Courses.** https://scholar.google.co.in/scholar?q=elective+recommendation+system+research+paper&hl=en&as_sdt=0&as_vis=1&oi=scholart#d=gs_qabs&t=1729009986929&u=%23p%3DZjIplrFp

[11] Sanjog Ray & Anuj Sharma (2011). **A Collaborative Filtering Based Approach for Recommending Elective Courses.** **https://link.springer.com/chapter/10.1007/978-3-642-194238_34?utm_source=researchgate.com**

[12] Chengyi Ju, Jiannong Cao, Yu Yang, Zhen-Qun Yang, & Ho Man Lee (2024). **Heterogeneity-aware Cross-school Electives Recommendation: a Hybrid Federated Approach.** **https://arxiv.org/abs/2402.12202?utm_source=publishcommunity.com**

[13] Muhammad Mukhtar Kurniawan & Aina Musdholifah (2021). **Elective Courses Recommendation System using Genetic Algorithm.**
https://dl.acm.org/doi/abs/10.1145/3489088.3489105

[14] Saurabh Amrutkar, Shantanu Mahakal, & Ajay Naidu (2021). **Recommender Systems for University Elective Course Recommendation.**
https://link.springer.com/chapter/10.1007/978-981-33-4862-2_27

APPENDIX-A

PSEUDOCODE

A. student_interface.py

```
# IMPORT NECESSARY LIBRARIES
IMPORT streamlit AS st
IMPORT pandas AS pd
IMPORT datetime
IMPORT functions FROM custom_modules

# DEFINE load_course_details FUNCTION
FUNCTION load_course_details(file_path):
    ACCEPT file_path AS input
    LOAD course details FROM file_path USING pandas.read_excel()
    RETURN course details

# DEFINE main FUNCTION
FUNCTION main():
    DISPLAY application title: "Elective Course Recommendation System"
    INITIALIZE database connection

    # LOAD AND PROCESS DATA
    LOAD grade data USING load_data FUNCTION
    LOAD course details USING load_course_details FUNCTION
    PREPROCESS grade data TO CREATE a pivot table
    COMPUTE similarity matrix FOR recommendation model

    # INITIALIZE SESSION STATE VARIABLES
    IF 'recommendations' NOT IN st.session_state:
        INITIALIZE st.session_state['recommendations'] AS an empty list

    # STUDENT ROLL NUMBER INPUT
    CREATE a text input field FOR student roll number USING st.text_input()
```

```
# VALIDATE ROLL NUMBER
IF a roll number IS entered:
    IF roll number NOT IN pivot table:
        DISPLAY error message: "Invalid roll number"
    ELSE:
        DISPLAY welcome message: "Welcome, [Student Name]!"

        # CHECK EXISTING ENROLLMENTS
        QUERY database TO CHECK IF student IS already enrolled:
            IF already enrolled:
                DISPLAY error message: "You are already enrolled in a course."
                RETURN
            ELSE:
                # GENERATE RECOMMENDATIONS
                IF st.session_state['recommendations'] IS EMPTY:
                    GENERATE recommendations USING recommend_courses
FUNCTION

                # DISPLAY RECOMMENDATIONS
                FOR each course IN st.session_state['recommendations']:
                    RETRIEVE course details AND enrollment count
                    IF seats ARE available:
                        DISPLAY course details (code, name, description, objective,
instructors, available seats)
                        CREATE "Enroll" button FOR course USING st.button()
                        IF button IS clicked:
                            ATTEMPT to enroll student IN course:
                                IF enrollment succeeds:
                                    DISPLAY success message: "Enrollment successful."
                                ELSE:
                                    DISPLAY error message: "Enrollment failed."
                    ELSE:
                        DISPLAY warning message: "Course is full."
```

```
DISPLAY separator BETWEEN courses

# RUN THE APPLICATION
IF __name__ == "__main__":
    EXECUTE main()
```

B. hod_interface.py

```
# IMPORT NECESSARY LIBRARIES
IMPORT streamlit AS st
IMPORT pandas AS pd
IMPORT matplotlib.pyplot AS plt
IMPORT seaborn AS sns
IMPORT functions FROM database_module

# DEFINE main FUNCTION
FUNCTION main():
    DISPLAY application title: "HOD Course Enrollment Dashboard"

    # FETCH ENROLLMENT AND COURSE DATA
    RETRIEVE all enrollments USING get_all_enrollments FUNCTION
    RETRIEVE list of all course codes USING get_all_courses FUNCTION

    # VISUALIZATIONS
    ## COURSE ENROLLMENT VISUALIZATION
    CREATE a bar chart TO show the number of students enrolled in each course:
        COMPUTE enrollment counts FOR each course
        PLOT bar chart USING seaborn's barplot FUNCTION
        LABEL axes, SET chart title, AND adjust rotation of x-axis labels FOR readability
        DISPLAY chart IN Streamlit

    ## COURSE ENROLLMENT PERCENTAGE VISUALIZATION
        CALCULATE enrollment percentage FOR each course BASED ON a fixed
      capacity of 60:
```

```
        COMPUTE percentage AS (enrolled_students / 60) * 100
        PLOT bar chart TO show enrollment percentage USING seaborn's barplot
        ANNOTATE chart WITH percentage values ABOVE each bar
        SET y-axis limits TO 0–100% AND adjust x-axis labels FOR readability
        DISPLAY chart IN Streamlit

    # STUDENT ENROLLMENT SECTION
    DISPLAY subheader: "Student Enrollments"
    ITERATE OVER all courses:
        DISPLAY course code AS a heading
        FILTER enrollments TO show only those FOR the current course

        ## SEARCH BOX FOR ROLL NUMBER
        CREATE a text input FOR searching roll numbers WITHIN the course enrollments
        IF search term IS entered:
            FILTER enrollments BASED ON the term

        ## DISPLAY STUDENT ENROLLMENTS
        IF enrollments EXIST FOR the course:
            DISPLAY columns FOR roll number, enrollment date, AND a delete button
            IF delete button IS clicked:
                REMOVE student's enrollment FROM database USING delete_enrollment
FUNCTION
                DISPLAY success message
                RELOAD page USING st.rerun()
        ELSE:
            DISPLAY message: "No enrollments found for this course."

        ## CSV DOWNLOAD OPTION
        PROVIDE a download button TO export the course's enrollments AS a CSV file:
            CONVERT enrollments TO CSV format
            ENABLE download USING Streamlit's download_button FUNCTION

# RUN THE APPLICATION
```

```
IF __name__ == "__main__":
    EXECUTE main()
```

C. recommendation_model.py

```
# DEFINE load_data FUNCTION
FUNCTION load_data(file_path):
    # Use pandas to read the Excel file
    DEFINE df AS pd.read_excel(file_path)
    # Print confirmation message
    PRINT "Data loaded successfully."
    # Return the loaded DataFrame
    RETURN df

# DEFINE preprocess_data FUNCTION
FUNCTION preprocess_data(df):
    # Create a pivot table
    DEFINE pivot_table AS pd.pivot_table(
        data=df,
        index='RollNumber',
        columns='CourseCode',
        values='Marks',
        fill_value=0
    )
    # Return the pivot table
    RETURN pivot_table

# DEFINE compute_similarity FUNCTION
FUNCTION compute_similarity(pivot_table):
    # Calculate cosine similarity for rows (students) of the pivot table
    FROM sklearn.metrics.pairwise IMPORT cosine_similarity
    DEFINE similarity_matrix AS cosine_similarity(pivot_table)
    # Return the similarity matrix
    RETURN similarity_matrix
```

```
# DEFINE recommend_courses FUNCTION
FUNCTION recommend_courses(student_id, pivot_table, similarity_matrix,
previous_recommendations=None, n_recommendations=5):
    # Check if student exists
    IF student_id NOT IN pivot_table.index:
        RETURN []

    # Find similar students
    DEFINE student_index AS pivot_table.index.get_loc(student_id)
    DEFINE similarity_scores AS similarity_matrix[student_index]
    DEFINE similar_students AS np.argsort(-similarity_scores)[1:11]  # Exclude the
student themselves

    # Generate course recommendations
    DEFINE recommendations AS {}
    FOR course IN pivot_table.columns:
        IF pivot_table.loc[student_id, course] == 0:  # Student has not taken the course
            IF previous_recommendations AND course IN previous_recommendations:
                CONTINUE
            # Compute the average score for the course among similar students
            DEFINE average_score AS np.mean([
                pivot_table.iloc[similar_student][course]
                FOR similar_student IN similar_students
            ])
            # Add course and average score to recommendations dictionary
            recommendations[course] = average_score

    # Select top recommendations
    DEFINE top_recommendations AS sorted(recommendations.items(), key=lambda
x: x[1], reverse=True)[:n_recommendations]
    DEFINE recommended_courses AS [course[0] FOR course IN
top_recommendations]
```

```
    # Return recommendations
    RETURN recommended_courses
```

D. system_performance.py

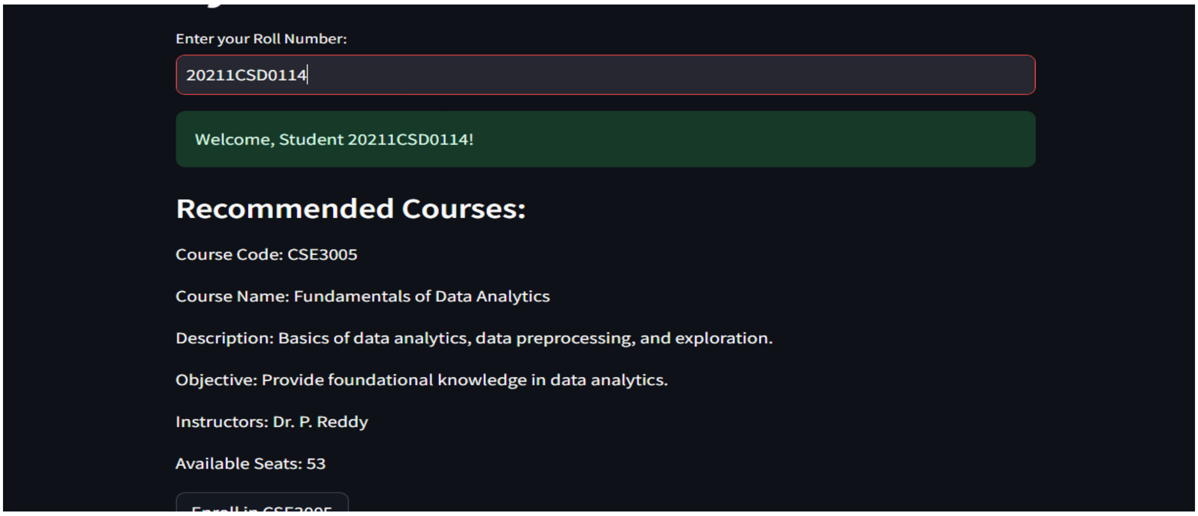
```
# DEFINE evaluate_recommendations FUNCTION
FUNCTION evaluate_recommendations(student_id, enrolled_courses,
recommendations):
    # Calculate hit count
    DEFINE hit_count AS len(set(enrolled_courses) & set(recommendations))

    # Calculate precision
    IF len(recommendations) > 0:
        DEFINE precision AS hit_count / len(recommendations)
    ELSE:
        DEFINE precision AS 0

    # Calculate recall
    IF len(enrolled_courses) > 0:
        DEFINE recall AS hit_count / len(enrolled_courses)
    ELSE:
        DEFINE recall AS 0

    # Calculate F1-score
    IF precision > 0 AND recall > 0:
        DEFINE F1 AS 2 * (precision * recall) / (precision + recall)
    ELSE:
        DEFINE F1 AS 0

    # Return precision, recall, and F1-score
    RETURN precision, recall, F1

# DEFINE main FUNCTION
FUNCTION main():
```

```
# Setup and Data Loading
DISPLAY application title: "Recommendation System Performance"
DEFINE grade_data AS load_data("grades.xlsx")
DEFINE pivot_table AS preprocess_data(grade_data)
DEFINE similarity_matrix AS compute_similarity(pivot_table)
DEFINE enrollments AS get_all_enrollments()

# Evaluate Performance for Each Student
DEFINE performance_data AS []
FOR student_id IN enrollments['StudentID'].unique():
    IF student_id IN pivot_table.index:
        DEFINE enrolled_courses AS enrollments[enrollments['StudentID'] ==
student_id]['CourseCode'].tolist()
        DEFINE recommendations AS recommend_courses(student_id, pivot_table,
similarity_matrix)
        DEFINE precision, recall, F1 AS evaluate_recommendations(student_id,
enrolled_courses, recommendations)
        APPEND {'StudentID': student_id, 'Precision': precision, 'Recall': recall, 'F1-
Score': F1} TO performance_data

DEFINE performance_df AS pd.DataFrame(performance_data)

# Display Overall System Performance
IF NOT performance_df.empty:
    # Display Key Metrics
    DISPLAY "Average Precision:", performance_df['Precision'].mean()
    DISPLAY "Average Recall:", performance_df['Recall'].mean()
    DISPLAY "Average F1-Score:", performance_df['F1-Score'].mean()

    # Visualize Performance Distributions
    DEFINE fig1, ax1 AS plt.subplots()
    sns.histplot(performance_df['Precision'], kde=True, ax=ax1)
    ax1.set_title("Precision Distribution")
    st.pyplot(fig1)
```

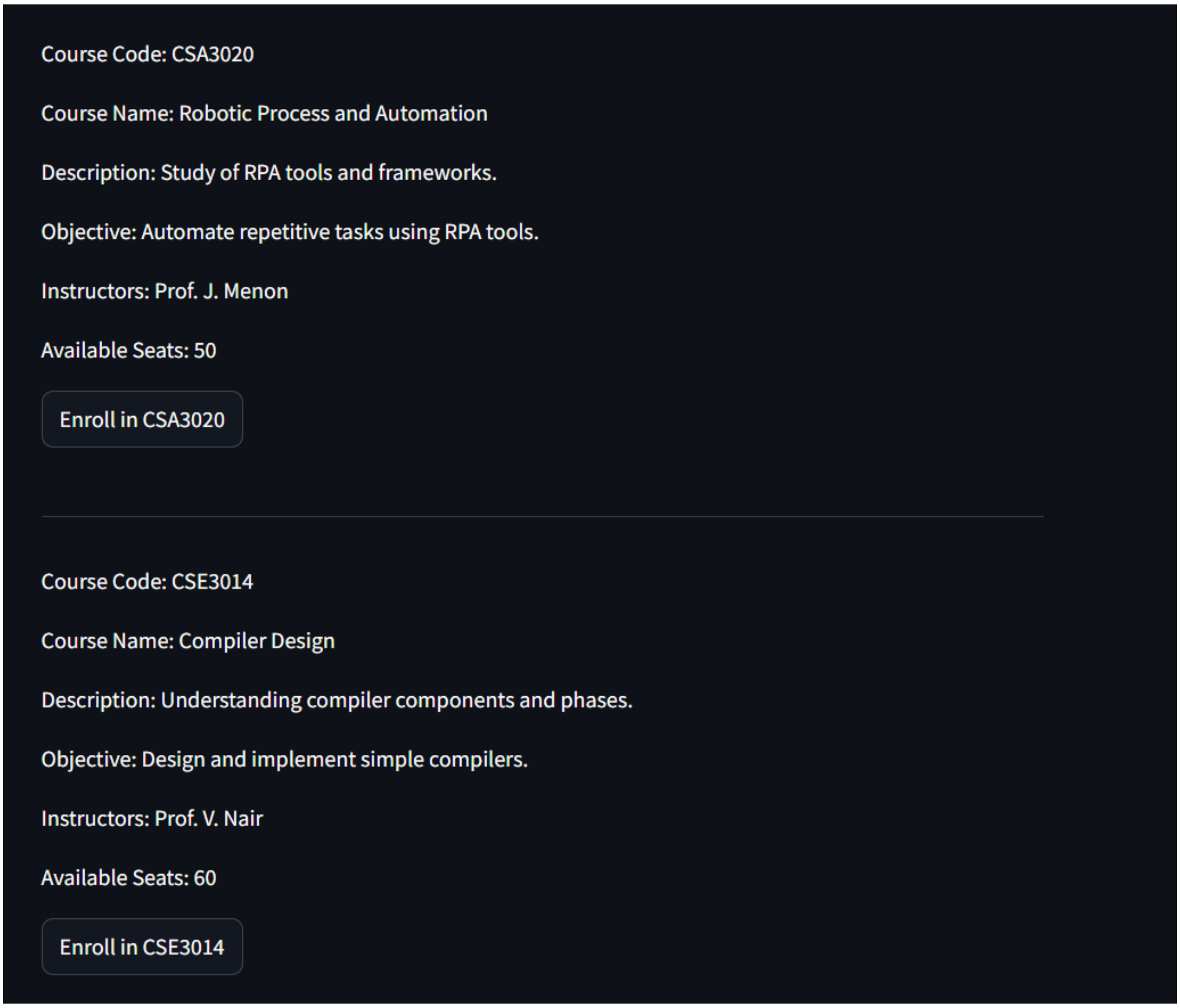
APPENDIX-B

SCREENSHOTS

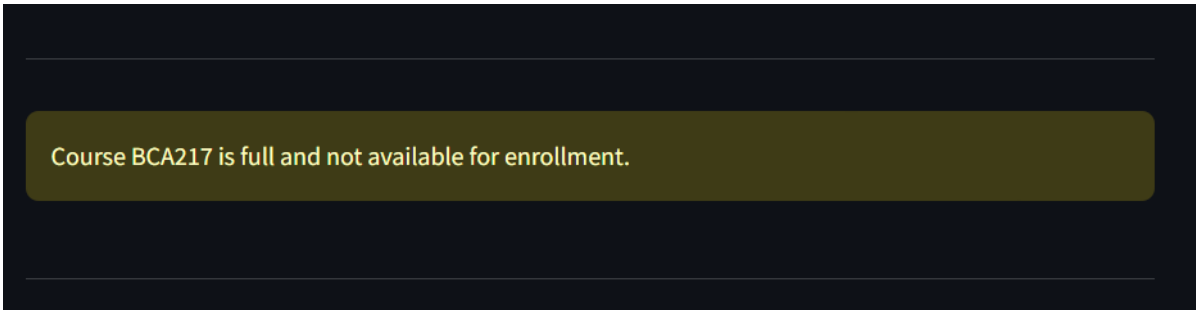
Screenshot 1: Streamlit app



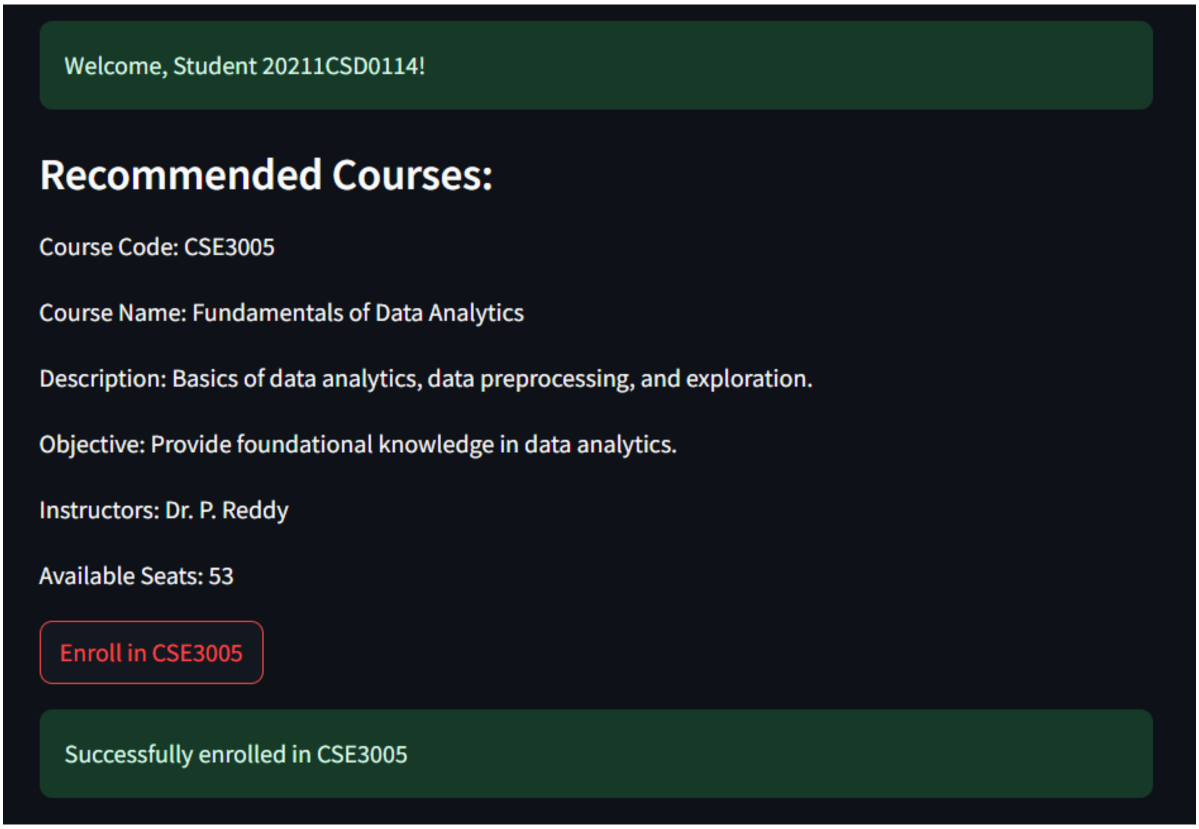
Screenshot 2: Recommendation



Screenshot 3: Recommendation

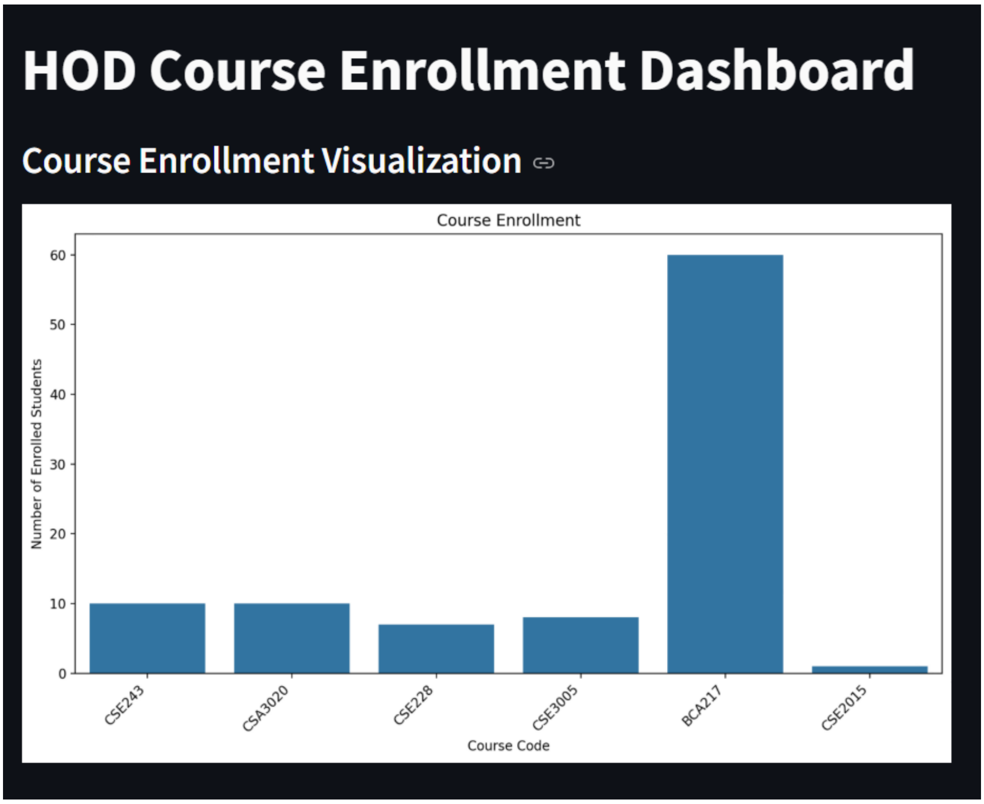


Screenshot 4: Unavailable course

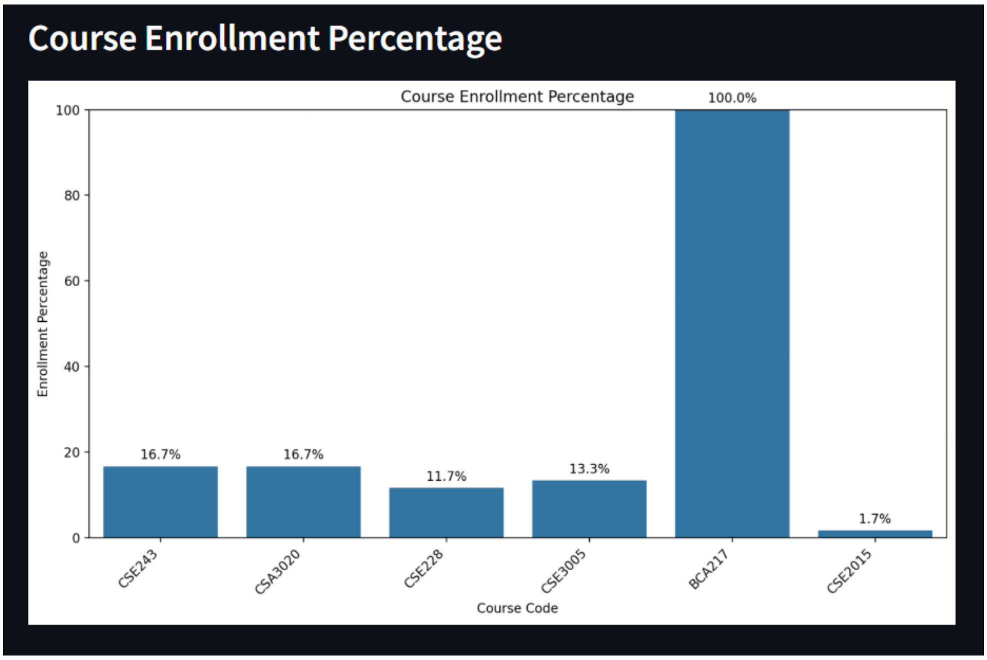


Screenshot 5: Enrollment

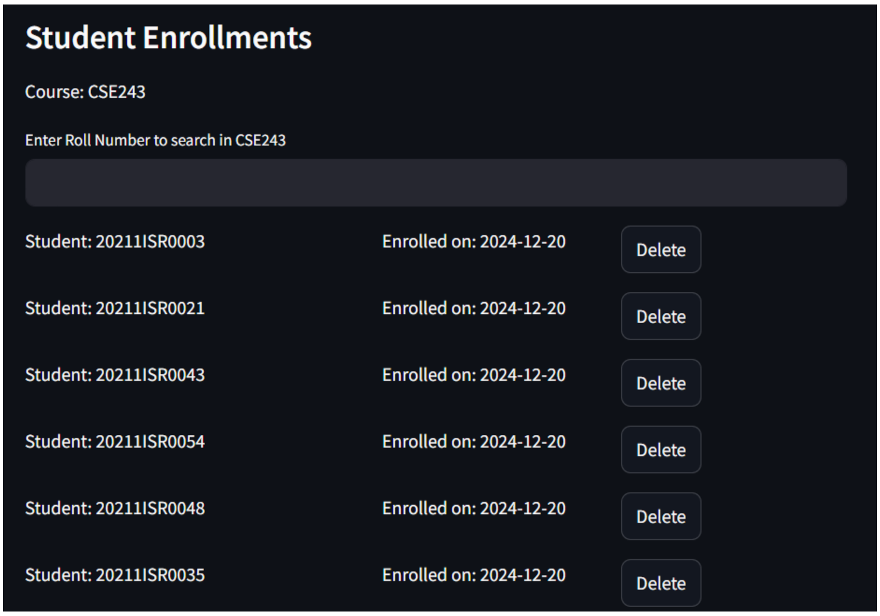
Screenshot 6: Existing Registration



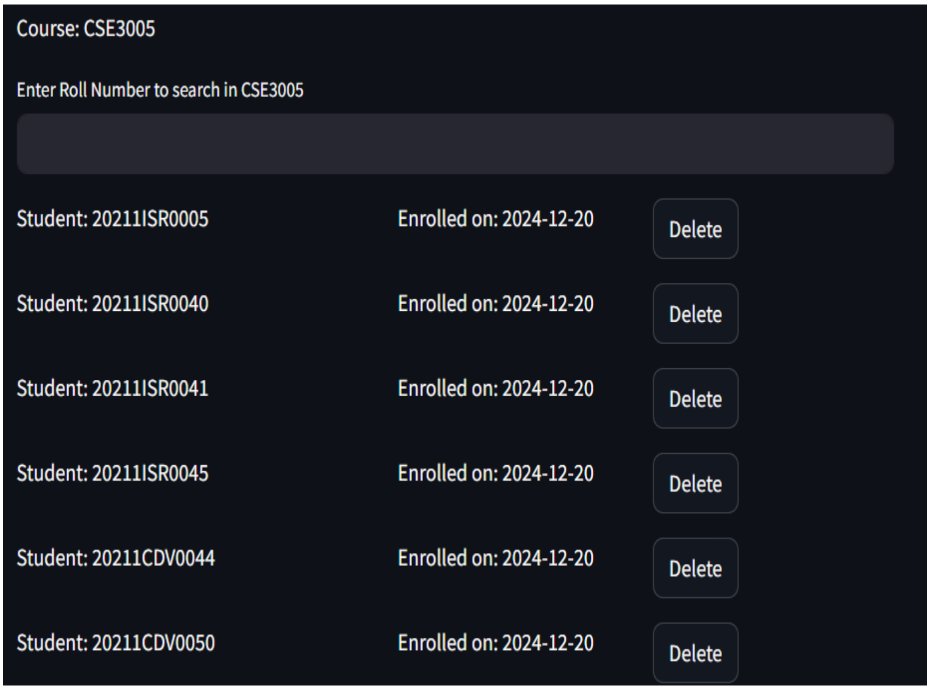
Screenshot 7: Course Enrollment Visualization



Screenshot 8: Course Enrollment Percentage



Screenshot 8: Student Enrollments



Screenshot 9: Delete Student Enrollment

APPENDIX-C

ENCLOSURES

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal Since 2013)

IJIRCCE

Impact Factor 8.625

CERTIFICATE

OF PUBLICATION

The Board of IJIRCCE is hereby awarding this certificate to

SAHANA R

UG Student, Dept. of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka, India

In Recognition of Publication of the Paper Entitled

"Elective Recommendation System using Machine Learning"

in IJIRCCE, Volume 13, Issue 1, January 2025

Google scholar Crossref Mendeley doi INNO SPACE SJIF Scientific Journal Impact Factor

e-ISSN: 2320-9801
p-ISSN: 2320-9798

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Editor-in-Chief

www.ijircce.com ijircce@gmail.com

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal Since 2013)

IJIRCCE

Impact Factor 8.625

CERTIFICATE

OF PUBLICATION

The Board of IJIRCCE is hereby awarding this certificate to

SRINIDHI S

UG Student, Dept. of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka, India

In Recognition of Publication of the Paper Entitled

"Elective Recommendation System using Machine Learning"

in IJIRCCE, Volume 13, Issue 1, January 2025

Google scholar Crossref Mendeley doi INNO SPACE SJIF Scientific Journal Impact Factor

e-ISSN: 2320-9801
p-ISSN: 2320-9798

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Editor-in-Chief

www.ijircce.com ijircce@gmail.com


International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal Since 2013)

IJIRCCE

Impact Factor 8.625

CERTIFICATE

OF PUBLICATION

The Board of IJIRCCE is hereby awarding this certificate to

AKASH KARTHIK RAO

UG Student, Dept. of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka, India

In Recognition of Publication of the Paper Entitled

"Elective Recommendation System using Machine Learning"

in IJIRCCE, Volume 13, Issue 1, January 2025

Google scholar · Crossref · Mendeley · doi · INNO SPACE SJIF Scientific Journal Impact Factor

e-ISSN: 2320-9801
p-ISSN: 2320-9798

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Editor-in-Chief

www.ijircce.com · ijircce@gmail.com

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal Since 2013)

IJIRCCE

Impact Factor 8.625

CERTIFICATE

OF PUBLICATION

The Board of IJIRCCE is hereby awarding this certificate to

PRATHIKSHA M

UG Student, Dept. of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka, India

In Recognition of Publication of the Paper Entitled

"Elective Recommendation System using Machine Learning"

in IJIRCCE, Volume 13, Issue 1, January 2025

Google scholar Crossref Mendeley doi INNO SPACE SJIF Scientific Journal Impact Factor

e-ISSN: 2320-9801
p-ISSN: 2320-9798

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Editor-in-Chief

www.ijircce.com ijircce@gmail.com


International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal Since 2013)

IJIRCCE

Impact Factor 8.625

CERTIFICATE

OF PUBLICATION

The Board of IJIRCCE is hereby awarding this certificate to

AMPANA J

UG Student, Dept. of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka, India

In Recognition of Publication of the Paper Entitled

"Elective Recommendation System using Machine Learning"

in IJIRCCE, Volume 13, Issue 1, January 2025

Google scholar Crossref Mendeley doi INNO SPACE SJIF Scientific Journal Impact Factor

e-ISSN: 2320-9801
p-ISSN: 2320-9798

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Editor-in-Chief

www.ijircce.com ijircce@gmail.com

K. MarimuthuReseach_paper.

ORIGINALITY REPORT

8%	5%	6%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Gyawali, Sujan. "Movie Recommendation System Using Machine Learning Techniques", Lamar University - Beaumont, 2023 Publication	1%
2	race.reva.edu.in Internet Source	1%
3	Submitted to Bournemouth University Student Paper	1%
4	Submitted to University of Surrey Student Paper	1%
5	Submitted to University of Ulster Student Paper	1%
6	Anandakumar H, Arulmurugan R. "Machine learning based Multi Agent Systems in Complex Networks", 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2019 Publication	1%
7	www.ijirset.com Internet Source	1%

SUSTAINABLE DEVELOPMENT GOAL MAPPING

The project worked carried out here is mapped to the following SDGs:

SDG 4: Quality Education

The project directly contributes to improving the quality of education by personalizing elective recommendations based on academic performance, interests, and career aspirations. This tailored approach ensures that students are engaged in relevant courses, enhancing their learning outcomes and satisfaction. Furthermore, the system fosters collaborative learning by grouping students with peers, creating a more inclusive and dynamic educational experience.

SDG 9: Industry, Innovation, and Infrastructure

By integrating advanced machine learning algorithms, such as collaborative filtering and hybrid models, the project promotes innovation in educational infrastructure. The system demonstrates the use of technology to automate complex processes, reduce administrative burdens, and enhance decision-making in academic institutions. The scalable design also ensures that the system can adapt to growing student populations and new electives, supporting sustainable educational development.

SDG 10: Reduced Inequalities

The system ensures equitable access to elective courses by implementing fair allocation mechanisms. Algorithms balance enrollments to prevent over-subscription or underutilization, ensuring that all students have equal opportunities to pursue their preferred electives. This approach reduces disparities in academic opportunities, particularly for students from diverse backgrounds.

SDG 17: Partnerships for the Goals

The project's framework, including its data-driven decision-making capabilities and integration of academic and social preferences, can be shared across institutions, fostering collaboration in education technology. By encouraging partnerships between universities and leveraging collective insights, the project aligns with global efforts to enhance academic planning and resource utilization.

By addressing these SDGs, the Elective Recommendation System demonstrates its potential as a transformative tool in education, enhancing accessibility, equity, and innovation in academic planning.