

Intelligent Course Recommender Portal for University Students Using Hybrid Collaborative Filtering

¹D. Jaya Soniya, ²Nalathoti Sai Harshitha, ³Kothuri Vasavi, ⁴Dova Archana, ⁵Myla Venkata Sowjanya

¹Assistant Professor, ^{2,3,4,5}Student,

Department of CSE – Data Science

St. Ann's College of Engineering and Technology, Chirala, Andhra Pradesh, India

Abstract—Selecting appropriate courses is a critical decision for university students, as it directly influences their academic performance and future career opportunities. Traditional course selection methods largely depend on informal guidance from peers or instructors, which may not provide accurate or personalized recommendations. To address this challenge, the present work proposes an Intelligent Course Recommender Portal that employs machine learning techniques to generate data-driven, personalized course suggestions. The system analyses historical academic data, including student grades and course enrolment patterns, to identify similarities and predict suitable courses. A hybrid collaborative filtering approach is implemented by combining user-based and item-based methods with an autoencoder-based model to capture complex non-linear patterns and improve prediction accuracy. Performance is evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The system is developed as a web-based application using Streamlit, providing an interactive interface for real-time recommendations along with predicted performance indicators. The results demonstrate that the proposed system improves the accuracy and reliability of course recommendations, thereby assisting students in making informed academic decisions.

Index Terms—Course Recommendation System, Collaborative Filtering, Autoencoder, Hybrid Filtering, Machine Learning, Streamlit, MAE, RMSE, Personalized Learning.

I. INTRODUCTION

In the modern digital era, intelligent systems have become essential for assisting users in making informed decisions based on large volumes of data. Recommendation systems, in particular, play a significant role in providing personalized suggestions by analysing user preferences, historical data, and behavioural patterns. These systems are widely applied in domains such as e-commerce, entertainment, and education to improve user experience and decision-making efficiency.

In academic environments, students are often required to select elective courses from a diverse set of available options. This process is crucial as it directly influences academic performance and future career opportunities. However, traditional course selection methods largely depend on informal guidance from peers or instructors, which may not always be reliable or personalized. This creates a need for a data-driven approach that can assist students in selecting suitable courses based on their academic history.

The proposed system introduces an Intelligent Course Recommender Portal that utilizes machine learning techniques to generate personalized course recommendations. The system analyses historical student-course data, including grades and enrolment patterns, to identify similarities among students and courses. Based on these similarities, the system predicts the most appropriate courses for each student.

The system employs collaborative filtering techniques, which are widely used in recommendation systems to predict user preferences based on the behaviour of similar users. Both user-based and item-based collaborative filtering methods are implemented to capture relationships between students and courses. In addition, autoencoder-based models are integrated to learn complex non-linear patterns in data and to improve prediction accuracy. Matrix factorization techniques further complement the framework by modelling latent relationships between users and items. By combining these approaches, the system achieves a hybrid recommendation model that enhances both accuracy and reliability.

The proposed system is developed as a web-based application using Streamlit, providing an interactive interface where students can input their details and receive real-time course recommendations. Along with the recommendations, the system provides predicted grade scores and confidence levels, enabling students to make well-informed academic decisions.

II. LITERATURE SURVEY

A. Introduction to Recommendation Systems

Recommendation systems are intelligent software systems designed to suggest relevant items to users based on their preferences and historical data. These systems are widely deployed in e-commerce, entertainment platforms, and educational settings to enhance user experience and decision-making. In academic environments, recommendation systems assist students in selecting suitable courses by analysing past performance and learning patterns. The fundamental idea is to predict user preferences for items not yet interacted with, achieved through collaborative filtering, content-based filtering, or hybrid approaches [1].

B. Collaborative Filtering Techniques

Collaborative filtering (CF) is a technique that makes predictions based on the behaviour and preferences of similar users. It assumes that users with similar past behaviour will have similar preferences in the future [2]. Two primary types of collaborative filtering methods are employed in the proposed system.

User-Based Collaborative Filtering: This method identifies users who have similar academic performance patterns and recommends courses based on the choices made by those similar students. Similarity between users is typically calculated using cosine similarity. While simple and effective, it may suffer from scalability issues when the number of users increases.

Item-Based Collaborative Filtering: This approach focuses on finding similarities between courses rather than students. Courses are recommended based on their similarity to courses in which the student has previously performed well. This method is more scalable than user-based filtering but may not capture complex relationships between students and items effectively.

C. Matrix Factorization Techniques

Matrix factorization is a powerful technique used in recommendation systems to discover latent features between users and items. It decomposes the user–item interaction matrix into lower-dimensional matrices that represent hidden relationships in the data [3]. This approach handles sparse datasets effectively and improves prediction accuracy by learning underlying patterns. Probabilistic matrix factorization further enhances this by incorporating probabilistic models to handle data uncertainty [4]. However, matrix factorization primarily captures linear relationships and may not fully model complex patterns in student behaviour.

D. Autoencoder-Based Recommendation Systems

Autoencoders are a type of neural network used for learning efficient representations of data. In recommendation systems, autoencoders reconstruct missing values in the user–item matrix by learning non-linear relationships between users and items [6]. The AutoRec model applies autoencoders for collaborative filtering tasks and has demonstrated superior performance compared to traditional methods, particularly in high-data-sparsity scenarios. Unlike conventional approaches, autoencoders capture complex patterns and dependencies, leading to improved recommendation accuracy.

E. Limitations of Existing Systems

Despite significant advancements, existing recommendation systems face several challenges. Data sparsity remains a critical issue, as student–course matrices often contain many missing values that affect prediction accuracy. The cold start problem presents difficulty in recommending courses for new students with no prior data. Many systems rely only on grades and ignore other factors such as student interests and career goals. Furthermore, traditional methods fail to capture complex non-linear relationships within the data, limiting their effectiveness.

Table I: Comparison of Existing Techniques

Method	Description	Advantages	Limitations
User-Based CF	Recommends based on similar users	Simple and intuitive	Scalability issues
Item-Based CF	Recommends based on similar items	Better scalability	Limited personalization
Matrix Factorization	Learns latent features from data	Handles sparsity well	Captures only linear relationships
Autoencoder	Uses neural networks for prediction	High accuracy, non-linear patterns	Requires more computational resources

III. PROPOSED METHODOLOGY

A. Proposed System

The proposed system is an Intelligent Course Recommender Portal designed to assist students in selecting suitable courses based on their academic performance and historical data. Unlike traditional methods that rely on manual guidance, this system uses machine learning techniques to generate personalized and data-driven recommendations. The system analyses student–course interaction data, including grades and course enrolment patterns, to identify similarities among students and courses. Based on these patterns, it predicts the most appropriate courses for each student along with expected performance outcomes.

The system adopts a hybrid collaborative filtering approach, combining both user-based and item-based techniques. An autoencoder-based model is further integrated to capture complex non-linear relationships within the data, thereby improving recommendation accuracy.

B. System Workflow

Data Collection: Student academic data, including course enrolments and grades, is collected and stored in a structured format.

Data Preprocessing: The collected data is cleaned and transformed into a student–course matrix for further analysis. Missing values are handled appropriately to ensure data quality.

Model Implementation: Different recommendation algorithms — user-based CF, item-based CF, and autoencoder — are applied to the processed matrix.

Prediction Generation: The system predicts expected grades for courses that the student has not yet taken.

Recommendation Output: Top-N recommended courses are presented alongside predicted grade scores and confidence levels through the web interface.

C. Performance Evaluation

The performance of the proposed system is evaluated using two standard metrics. Mean Absolute Error (MAE) measures the average difference between predicted and actual grade values. Root Mean Squared Error (RMSE) penalizes larger errors more strongly by computing the square root of the mean squared differences. These metrics provide a comprehensive measure of how reliably the system predicts student performance. Based on comparative analysis, the autoencoder-based model achieved the lowest error rates among all methods evaluated, followed by user-based collaborative filtering, with matrix factorization showing moderate performance.

Table II: Comparison with Existing System

Feature	Existing System	Proposed System
Recommendation Method	Manual / basic filtering	Hybrid collaborative filtering
Accuracy	Low to moderate	High — autoencoder-enhanced
Personalization	Limited	Highly personalized
Data Usage	Minimal	Historical academic data
Decision Support	Not reliable	Data-driven recommendations

Table III: Key Observations

Observation	Explanation
Autoencoders perform best	Capture complex non-linear patterns in student data
User-based > Item-based	Student similarity is a stronger predictor
Hybrid model improves accuracy	Combines strengths of multiple methods
80:20 train-test split	Prevents overfitting; ensures generalization

IV. SYSTEM DESIGN

A. Architecture of the System

The architecture of the Intelligent Course Recommender Portal is structured in a modular manner, where each component performs a specific function in the recommendation pipeline. The main components are as follows.

Data Collection Module: Collects student academic data such as course history and grades from structured data files.

Data Preprocessing Module: Cleans and transforms data into a structured student–course matrix suitable for model training.

Recommendation Engine: Implements user-based CF, item-based CF, and autoencoder algorithms to identify the most suitable courses.

Prediction Module: Predicts expected grades for courses not yet taken by a student, ranking them by predicted performance.

User Interface Module: Provides an interactive web interface using Streamlit for real-time recommendation display.

This layered architecture ensures smooth data flow from input to output, enabling efficient processing and accurate predictions.

B. Dataset Structure

The student–course dataset forms the foundation of the recommendation system. It is structured as a matrix where rows represent students and columns represent courses, with grade values as entries. Missing values correspond to courses not yet taken and serve as prediction targets for the recommendation models.

Table IV: Student-Course Dataset Structure

Field Name	Description
Student ID	Unique identifier for each student
Course Code	Unique identifier for each course
Course Name	Name of the course
Grade	Grade obtained by the student
Semester	Semester in which the course was taken

C. UML Diagrams

The system design is further represented through Unified Modelling Language (UML) diagrams that capture the structure and behaviour of the system. The class diagram includes entities such as Student, Course, StudentCourse, RecommendationModel, and Recommendation, illustrating how these components interact. The use case diagram depicts the student as the primary actor who inputs a student ID and semester, receives course recommendations, and views predicted grades and evaluation metrics. The sequence diagram illustrates the interaction flow between the student, web interface, recommendation engine, and model storage. The state chart diagram represents the lifecycle of the system from input collection to final recommendation display.

V. IMPLEMENTATION

A. Software Requirements

The system is implemented using Python 3.8 or higher, utilizing machine learning libraries including NumPy, Pandas, Scikit-learn, and TensorFlow/Keras. The frontend is developed using Streamlit, which enables seamless integration of machine learning models with a web-based interface. Development is performed using VS Code or PyCharm on Windows 10/11, macOS, or Linux operating systems. The application is accessible through standard web browsers such as Google Chrome or Firefox.

B. Hardware Requirements

The system requires moderate computational resources. A processor of Intel Core i3 or higher, a minimum of 4 GB RAM (8 GB recommended), and at least 20 GB of free storage on a 64-bit architecture system are sufficient to run the recommendation models and web application smoothly.

C. Recommendation Algorithms

User-Based Collaborative Filtering: This algorithm predicts the grade of a student in a particular course by analysing the performance of similar students. Cosine similarity is used to measure student proximity, and predictions are generated using a weighted average of neighbouring students' grades [1][2].

Item-Based Collaborative Filtering: This algorithm predicts grades by analysing the relationship between courses. It identifies courses similar to those already taken by the student and estimates performance based on prior grades [2].

Autoencoder-Based Model (AutoRec): The autoencoder model is a neural network-based approach that learns latent representations from the student-course matrix. It reconstructs missing grade values by minimizing reconstruction error and captures complex patterns in student performance data. This model provides better accuracy compared to traditional collaborative filtering methods due to its ability to learn non-linear relationships [6].

D. Key Implementation Modules

The Data Processing Module loads and preprocesses the student-course dataset from structured pickle files, including the training matrix, course-to-index mappings, course names, semester mappings, and student data. The Model Training Module splits the dataset using an 80:20 train-test ratio, trains the collaborative filtering and autoencoder models, and evaluates performance using MAE and RMSE. The Prediction Module accepts a student ID as input, retrieves academic history, predicts grades for unseen courses using the selected trained model, ranks them by predicted score, and returns the top-N recommended courses. The User Interface Module provides a Streamlit-based dashboard with navigation options including Problem Statement, Dataset Description, Methodology, Results and Performance, and Get Recommendations.

VI. TESTING AND VALIDATION

Testing is an essential phase in the development of the Intelligent Course Recommender Portal, ensuring that the system performs accurately, efficiently, and reliably under different conditions. Functional testing verifies that the system generates appropriate course recommendations based on user input. Unit testing validates individual modules such as data processing, model prediction, and the user interface. Performance testing evaluates the response time and efficiency of the recommendation pipeline.

Table V: Test Cases and Expected Outcomes

Test Case ID	Test Scenario	Expected Output	Status
TC-01	User selects a valid Student ID	Student ID displayed correctly	Pass
TC-02	User requests course recommendations	Top recommended courses displayed	Pass
TC-03	Autoencoder model selection	Recommendations with improved accuracy	Pass
TC-04	Switching between algorithms	Recommendations updated accordingly	Pass
TC-05	Invalid or missing input	Error handled gracefully	Pass
TC-06	Performance testing	Recommendations generated in acceptable time	Pass
TC-07	Data loading and model execution	Data and models loaded without failure	Pass
TC-08	UI interaction	User navigates features smoothly	Pass

The testing results indicate that the system performs efficiently across all modules. The recommendation algorithms successfully generate relevant course suggestions. The autoencoder-based model demonstrates improved accuracy compared to traditional collaborative filtering techniques. Performance testing confirms that the system generates recommendations within a short response time, and evaluation metrics including MAE and RMSE confirm the reliability of predictions.

VII. RESULTS AND DISCUSSION

The results of the proposed system demonstrate the effectiveness of hybrid collaborative filtering combined with deep learning in generating accurate course recommendations. The system provides an interactive, web-based dashboard through which students can explore the methodology, view dataset descriptions, and obtain personalized recommendations.

Model training and evaluation were performed using the student–course matrix with an 80:20 train-test split. A comparison of model performance indicates that the autoencoder-based model achieves the best results with the lowest MAE and RMSE values among all methods evaluated. User-based collaborative filtering performs better than item-based collaborative filtering, as student similarity is a more meaningful predictor than course-to-course similarity. Item-based CF and Logistic MF show moderate results, while WARP MF serves as a baseline comparator with relatively higher error metrics.

The recommendation output for a selected student ID includes the top five recommended courses for a specified target semester. For each course, the system presents the course code, predicted grade score, expected performance category, letter grade, and confidence level. This multi-dimensional output enables students to evaluate their likely performance in each recommended course before making a selection, thereby facilitating informed academic planning.



Course Code	Course Name	Predicted Grade	Letter Grade	Confidence
CSE400	CSE400- Advanced Topics	10.00	A	0.26
CSE401	CSE401- Advanced Topics	10.00	A	0.25
CSE404	CSE404- Advanced Topics	10.00	A	0.22
CSE408	CSE408- Advanced Topics	10.00	A	0.33
CSE410	CSE410- Advanced Topics	10.00	A	0.22

Figure 1 showing recommendation summary for courses

The experimental observations confirm that the hybrid model combining collaborative filtering and autoencoder-based techniques delivers more reliable recommendations than any single method applied independently. The 80:20 data split is found to prevent overfitting and ensure generalization across unseen student data.

VIII. CONCLUSION

The proposed Intelligent Course Recommender Portal successfully demonstrates the application of machine learning techniques in assisting university students with course selection. By utilizing historical academic data, the system provides personalized recommendations that help students make informed decisions. The implementation of collaborative filtering methods, including user-based and item-based approaches, enables the system to identify similarities between students and courses effectively.

The integration of an autoencoder-based model enhances prediction accuracy by capturing complex non-linear relationships within the data. Evaluation results measured using MAE and RMSE indicate that the autoencoder model outperforms traditional methods in generating reliable recommendations. The system is developed as a web-based application using Streamlit, offering an interactive and user-friendly interface for accessing recommendations in real time.

Overall, the project provides a scalable and efficient solution for course recommendation, reducing uncertainty in course selection and improving academic planning for students. The system is designed to be adaptable to diverse academic environments, making it a practical tool for data-driven educational decision support.

IX. FUTURE SCOPE

The proposed system can be further enhanced by incorporating additional features and improvements to increase its accuracy, scalability, and usability. Currently, the system primarily relies on student grades and course interaction data. Future work may integrate additional features such as student interests, career goals, course difficulty levels, and prerequisites to provide more personalized recommendations.

Potential enhancements include the integration of advanced deep neural network architectures and hybrid models combining collaborative and content-based filtering, as well as explainable AI mechanisms that provide clear reasoning for each recommended course to increase transparency and user trust. Deployment as a scalable web and mobile application would allow access to a larger user base with real-time recommendations. Integration with university databases would enable automatic data updates and real-time academic tracking. Multi-semester planning capability, where the system suggests a sequence of courses for long-term academic planning, represents a significant future direction that would transform the system into a comprehensive academic decision-support tool.

ACKNOWLEDGMENT

The authors express sincere gratitude to their guide, D. Jaya Soniya, Assistant Professor, and the Head of the Department of CSE – Data Science, Dr. K. Subbarao, St. Ann's College of Engineering and Technology, Chirala, for their valuable guidance and support throughout this project. The authors also acknowledge the teaching and non-teaching staff of the Department of CSE – Data Science for their cooperation, and the management of St. Ann's College of Engineering & Technology for providing a conducive research environment and excellent laboratory facilities.

REFERENCES

- [1] M. Papagelis and D. Plexousakis, "Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents," *Engineering Applications of Artificial Intelligence*, vol. 18, no. 7, pp. 781-789, 2005.
- [2] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "An algorithmic framework for performing collaborative filtering," *Proceedings of the ACM SIGIR Conference*, pp. 158-165, 1999.
- [3] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, pp. 30-37, 2009.
- [4] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," *Advances in Neural Information Processing Systems*, 2008.
- [5] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," *Proceedings of WWW*, 2017.
- [6] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "AutoRec: Autoencoders meet collaborative filtering," *Proceedings of WWW*, 2015.
- [7] J. Weston, S. Bengio, and N. Usunier, "WSABIE: Scaling up to large vocabulary image annotation," *Proceedings of IJCAI*, 2011.
- [8] S. Rendle, "Factorization machines," *Proceedings of ICDM*, 2010.
- [9] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, 2002.
- [10] N. Thai-Nghe, L. Drumond, and L. Schmidt-Thieme, "Recommender system for predicting student performance," *Procedia Computer Science*, 2010.
- [11] M. Kula, "LightFM: A hybrid recommendation model," *LightFM Documentation*, 2015.
- [12] M. Kula, "Weighted Approximate-Rank Pairwise (WARP) loss," *LightFM Documentation*, 2015.
- [13] S. Arya and S. Dhawan, "Course Recommendation System," *Indraprastha Institute of Information Technology Delhi*, 2018.
- [14] H. Steck, "AutoRec: Autoencoders meet collaborative filtering," *Proceedings of the 24th International Conference on World Wide Web (WWW)*, 2015.