AI POWERED COURSE SELECTOR

Priyanka Manke[1]; Vedant Suralkar[2]; Yash Maismale[3]; Mahesh Sabale[4]; Saniya Ansari[5].
Vasantdada Patil Pratishthan's College of Engineering and V.A. Dept. of Information Technology, Mumbai, India.

ABSTRACT:

This project explores six methodologies for enhancing course recommendation systems, focusing on personalized learning and academic success. It improves collaborative filtering by analyzing implicit user behavior and employs deep learning techniques like CNNs for personalized recommendations. Predictive modeling methods, including k-nearest neighbors and matrix factorization, help forecast student performance. A skill-based system uses fuzzy clustering to align courses with career goals, while a hybrid approach integrates clustering and association rule mining to generate recommendations based on course history and grades. By combining these techniques, the system aims to optimize accuracy and provide tailored course suggestions.

Keywords: Collaborative filtering, deep learning, fuzzy clustering, matrix factorization, user implicit behavior analysis, sequential pattern mining, academic performance prediction.

I. INTRODUCTION:

In today's educational landscape, students struggle to choose the right courses due to diverse preferences and goals. Course recommendation systems provide personalized suggestions based on interests, past performance, and learning behavior. Using collaborative filtering and deep learning, these systems analyze large datasets to improve accuracy. Collaborative filtering identifies similar students, while CNNs detect complex patterns. Clustering and association rule mining refine recommendations by grouping students and uncovering trends. These methods tackle challenges like data sparsity and real-time adaptability. This project integrates these techniques to develop an efficient system that enhances student decision-making and academic success. system that enhances student decision-making and academic success.

II. LITERATURE SURVEY:

The proposed solution includes an item-based collaborative filtering model enhanced with user implicit behavior data, such as login, learning, and course selection details, to address the challenge of data sparsity. Historical preference fusion is introduced to mitigate cold-start issues by grouping users with similar interests, which improves the model's overall performance. The model's precision and recall are validated under varying K-values using real-world education platform data (1).Additionally, a user-based collaborative filtering system is designed, comprising a five-step workflow: data collection, ETL, model generation, strategy configuration, and service supply. Pearson correlation is employed to compute learner similarity, while association rules are used for top-N recommendations. Non-personalized recommendations are highlighted as a baseline for ephemeral user sessions, ensuring better initial user experiences (2).A CNN-based recommendation system for textile patterns is also developed, focusing on color compatibility using a custom dataset of 12,000 images. The system classifies patterns into five categories based on user and designer surveys, and integrates periodic user feedback for model retraining. With an accuracy of 82.08%, the model demonstrates the viability of deep learning in non-traditional recommendation domains (3).The system compares k-NN, matrix factorization (MF), and biased MF for predicting student performance, treating students as users and courses as items. Biased MF outperforms other methods with an RMSE of 0.831 by incorporating student/course bias terms. A framework is proposed that integrates both training (desktop) and recommendation (web) modules, enhancing scalability (4).Fuzzy c-means clustering is applied to group students based on domain interests, effectively addressing the vagueness in human-related data. The introduction of a weighted_mode metric helps prioritize courses taken by clustered students, achieving 81.3% accuracy with real data. The model recommends courses aligned with job domains, such as "Machine Learning with Python," using alumni data (5).Lastly, k-means clustering and Apriori/SPADE algorithms are used to generate course association rules, revealing higher rule coverage in clustered datasets. Sequential pattern mining through SPADE is employed to model course dependencies and prerequisite relationships, enabling course recommendations based on minimum acceptable grades and historical sequences of high-performing students (6).

III. METHODOLOGY:

Data collection involves gathering historical data from educational platforms, including student profiles, course enrollments, grades, and implicit behaviors like login frequency and course interactions. **Data preprocessing** ensures data quality by handling missing values, normalizing grades, and removing outliers. **Feature extraction** identifies relevant aspects such as student interests, course prerequisites, and performance metrics. A robust dataset is built using these extracted features. **Clustering** is then applied to group students based on similarities. The fuzzy c-means clustering method is used for flexibility in assignments. This approach helps in understanding student behavior and performance. It also aids in personalized recommendations. Ultimately, it enhances the overall learning experience. here are three types of algorithms used for course recommendations: Collaborative Filtering, Matrix Factorization, and Association Rule Mining.

A. *Collaborative Filtering*:

Implement item-based collaborative filtering to recommend courses based on similarities between courses taken by students.

B. *Matrix Factorization*:

Use biased matrix factorization to predict student performance by incorporating student and course biases, improving recommendation accuracy.

C. *Association Rule Mining*:

Apply the Apriori algorithm to generate association rules from clustered data, identifying frequent course patterns and dependencies.

D. *Sequential Pattern Mining*:

Use the SPADE algorithm to model course sequences and dependencies, ensuring recommendations align with prerequisite structures.

Weighted Mode Calculation prioritizes courses taken by students in similar clusters using a weighted_mode metric to enhance recommendation relevance. **Model Training** employs 5-fold cross-validation for robustness and generalizability, while **User Feedback** Integration ensures models adapt to changing preferences by periodically retraining with feedback. Recommendation Generation provides personalized course suggestions based on predicted grades, student interests, and course dependencies.**System Architecture**, User Interface Development, and Deployment and Testing together ensure a modular design, intuitive web-based access, and real-world validation for effectiveness and usability.

IV. PROPOSED SYSTEM

System Architecture

The proposed system is designed with a modular architecture consisting of key components. The **Data Collection Module** gathers student profiles, course enrollments, grades, and behavioral data like login frequency. The **Preprocessing Module** cleans, normalizes, and extracts important features such as interests, prerequisites, and performance trends. The **Clustering Module** applies fuzzy c-means clustering to group students based on academic history and interests. The **Model Training Module** includes Collaborative Filtering for course similarity, Matrix Factorization for grade prediction, and CNNs for behavioral pattern analysis. The **Rule Generation Module** mines association rules and sequential patterns using Apriori and SPADE algorithms. The **Recommendation Engine** integrates model predictions and rule-based insights to generate personalized course suggestions. The **User Interface** provides a web-based platform where students can view recommendations, set grade thresholds, and provide feedback. The **Data Input Stage** collects historical and real-time student data for analysis. The **Preprocessing Stage** ensures high-quality input by normalizing and encoding data while removing noise. The **Recommendation Generation Stage** filters courses based on prerequisites and grade thresholds.

Key Algorithms and Techniques

A. Fuzzy C-Means Clustering: Groups students into clusters with membership degrees (Fig. 3).
   Input: Student-course matrix.
   Output: Clusters with weighted memberships.
B. Apriori Algorithm: Mines frequent course sets (e.g., *{DBMS, Algorithms}* → *{ML}*).
C. SPADE Algorithm: Identifies sequential dependencies (e.g., *Programming* → *Data Structures*).

System Evaluation

A. Metrics:
   Precision/Recall: Measure recommendation relevance.
   RMSE: Evaluate grade prediction accuracy.
   Coverage: Assess rule applicability across clusters.
B. Validation: 5-fold cross-validation on historical data.
C. User Testing: Collect feedback on recommendation quality via surveys.

Innovation and Impact

Hybrid Approach: Combines collaborative filtering, deep learning, and rule mining for robust recommendations.
Dynamic Adaptation: Retrains models using user feedback to address concept drift.
Job Alignment: Recommends courses aligned with alumni career paths (e.g., "Machine Learning with Python" for ML roles).

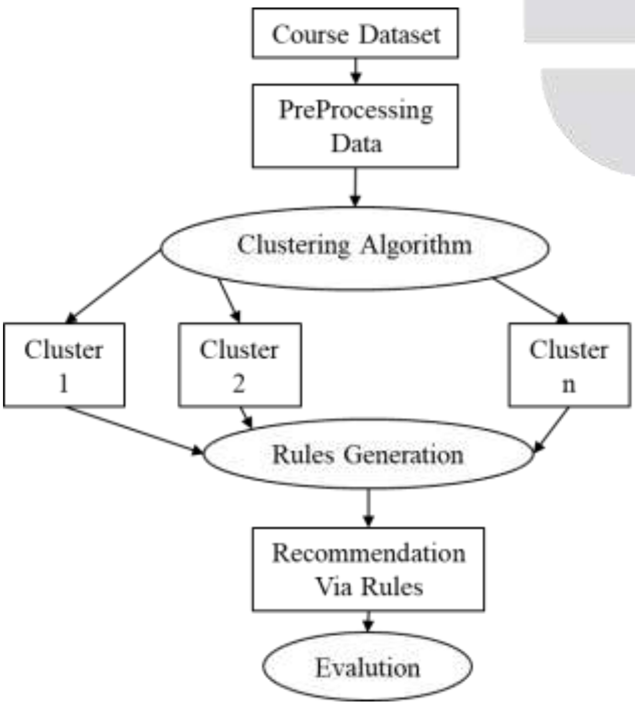


Fig. 1 System Flowchart

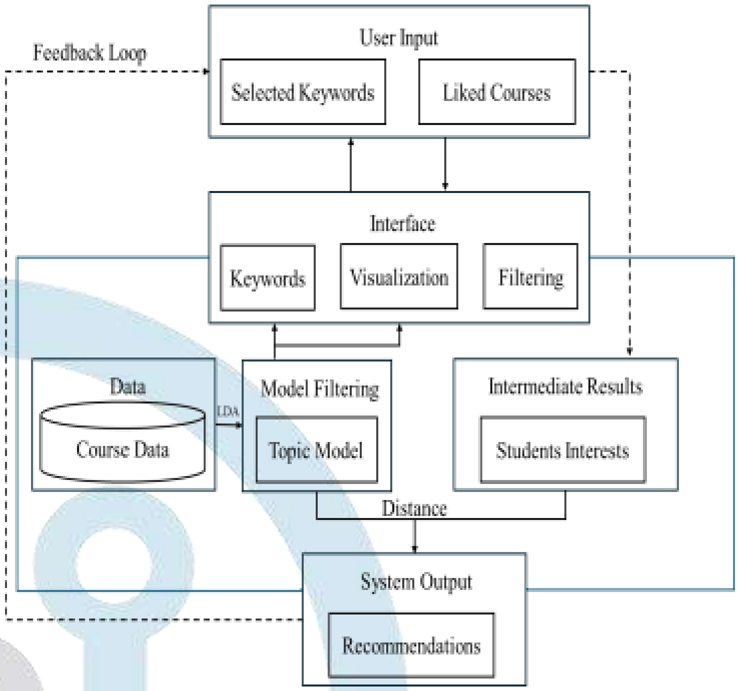


Fig. 2 System Architecture

V. SOFTWARE USED

A. Programming Languages: Python: Primary language for data preprocessing, model training (collaborative filtering, matrix factorization, CNN), and web development. **R**: Used for implementing the **SPADE algorithm** (sequential pattern mining) and association rule mining.

B. Development Environments: PyCharm: For Python scripting and model development (Paper 1). **Jupyter Notebook**: For exploratory data analysis and prototyping. **RStudio**: For R-based tasks like SPADE and Apriori implementations (Paper 6).

C. Machine Learning & Data Mining Libraries: Scikit-learn: For clustering (k-means, fuzzy c-means via skfuzzy) and preprocessing. **TensorFlow/Keras**: For building and training the **CNN-based deep learning model** (Paper 3). **Surprise**: Python library for collaborative filtering and matrix factorization. **mlxtend**: For association rule mining (Apriori algorithm). **Weka**: Java-based tool for clustering and data preprocessing (Paper 4).

D. Database Management: MySQL/PostgreSQL: For storing student profiles, course data, and recommendation rules.

E. Web Development Frameworks:
Flask: Backend frameworks for building the web-based user interface.
HTML/CSS/JavaScript: Frontend technologies for designing interactive dashboards.

F. Version Control:
Git/GitHub: For collaborative development and code management.

G. Visualization Tools:
Matplotlib/Seaborn: For plotting precision-recall curves, RMSE trends, and cluster visualizations. **Tableau**: Optional for advanced dashboarding (implied but not explicitly mentioned).

VI. FUTURE SCOPE

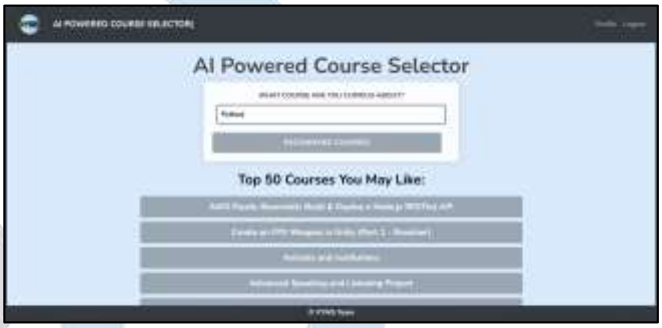
1. **Integration with MOOCs**: Expand the system to recommend courses from platforms like Coursera, edX, and Udemy.
2. **Real-Time Adaptation**: Implement real-time updates using streaming data to adapt recommendations dynamically.
3. **Multi-Domain Recommendations**: Extend the system to recommend courses across multiple domains (e.g., business, arts, sciences).
4. **Career Path Alignment**: Incorporate job market trends and alumni career data to align recommendations with industry demands.
5. **Gamification**: Add gamified elements (e.g., badges, leaderboards) to encourage course completion and engagement.
6. **Collaboration with Universities**: Partner with universities to integrate the system into their academic advising processes.

VII. Results

A. Registration Page

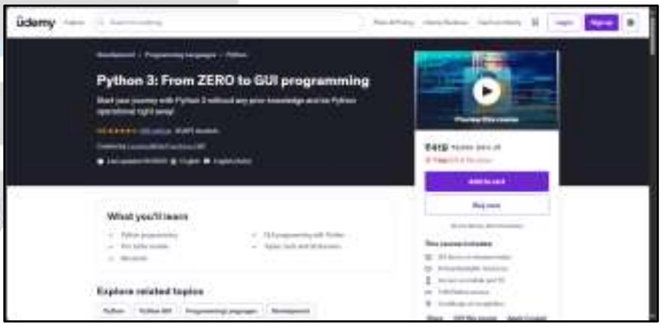
B. Login Page

C. Main Screen



D. Recommended Courses

E. Selected Course



VIII. CONCLUSION

The course recommendation system uses collaborative filtering, deep learning, and data mining for personalized suggestions. It tackles challenges like data sparsity, cold-start issues, and course dependencies. A modular, hybrid approach ensures scalability and adaptability. Future upgrades include real-time updates and career alignment. This enhances accuracy and relevance in recommendations. The system empowers students to make informed academic and career choices.

REFERENCE

[1]. Jena, K. K., Bhoi, S. K., Malik, T. K., Sahoo, K. S., Jhanjhi, N. Z., Bhatia, S., & Amsaad, F. (2023). E-Learning Course Recommender System Using Collaborative Filtering Models. Electronics, 12(1), 157. https://doi.org/10.3390/electronics12010157

[2]. Sankhe, V., Shah, J., Paranjape, T., & Shankarmani, R. (2020). Skill Based Course Recommendation System. 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON). doi:10.1109/gucon48875.2020.923107

[3]. Thanh-Nhan, H.-L., Huu-Hoa, N., & Thai-Nghe, N. (2016). Methods for Building Course Recommendation Systems. *2016 Eighth International Conference on Knowledge and Systems Engineering (KSE)*, 163-168. https://doi.org/10.1109/KSE.2016.7758051

[4]. Reisoğlu, Ü. T., Tuncer, A., Savran, H., & Yılmaz, S. (2020). An Online Recommendation System Using Deep Learning for Textile Products. *2020 International Conference on Computer Science and Software Engineering*, 1-4. https://doi.org/10.1109/CSSE.2020.305

[5]. Tan, H., Guo, J., & Li, Y. (2008). E-Learning Recommendation System. *2008 International Conference on Computer Science and Software Engineering*, 430-433. https://doi.org/10.1109/CSSE.2008.305

[6]. Zhao, L., & Pan, Z. (2021). Research on Online Course Recommendation Model Based on Improved Collaborative Filtering Algorithm. *2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, 437-440. https://doi.org/10.1109/ICCCBDA51879.2021.9442575

[7]. Fanca, A., Gota, D.-I., Puscasiu, A., & Valean, H. (2020). Recommendation Systems with Machine Learning. *2020 International Conference on Computer Science and Software Engineering*, 1-4. https://doi.org/10.1109/CSSE.2020.305

[8]. Yanes, N., Mostafa, A. M., Ezz, M., & Almuayqil, S. N. (2020). A Machine Learning-Based Recommender System for Improving Students Learning Experiences. *IEEE Access, 8*, 201218–201233. https://doi.org/10.1109/ACCESS.2020.3036336

[9]. Kommineni, M., Alekhya, P., Vyshnavi, T. M., Aparna, V., Swetha, K., & Mounika, V. (2020). Machine Learning based Efficient Recommendation System for Book Selection using User based Collaborative Filtering Algorithm. 2020 Fourth International Conference on Inventive Systems and Control (ICISC). doi:10.1109/icisc47916.2020.9171222

[10]. Z. Ibraimi, M. Mehmeti and S. G. Mahmutbegovic, "Artificial Intelligence in Traffic Management Systems: A Comprehensive Review," 2022 21st International Symposium INFOTEH-JAHORINA (INFOTEH), 2022, pp. 1-6, doi: 10.1109/INFOTEH53737.2022.9751282.