



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

# IJRSET

International Journal of Innovative Research in  
**SCIENCE | ENGINEERING | TECHNOLOGY**

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 5, May 2023

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

[ijrset@gmail.com](mailto:ijrset@gmail.com)

[www.ijrset.com](http://www.ijrset.com)



# Enhancing E-Learning Experiences: A Novel Approach to Personalized Course Recommendations

Prajwal Waykos<sup>1</sup>, Dhairya Thakkar<sup>2</sup>, Soham Tolwala<sup>3</sup>, Pushpak Suryawanshi<sup>4</sup>,  
Ass.Prof. Gitanjali Yadav<sup>5</sup>

U.G. Student, Department of Artificial Intelligence and Data Science, Vishwakarma Institute of Information  
Technology, Kondwa Bk., Pune, India<sup>1-4</sup>

Assistant Professor, Department of Artificial Intelligence and Data Science, Vishwakarma Institute of Information  
Technology, Kondwa Bk., Pune, India<sup>5</sup>

**ABSTRACT:** With the proliferation of e-Learning platforms, the availability of numerous courses poses a challenge for users in selecting the most suitable ones. Recommender systems have emerged as a crucial solution to this problem by automating the course selection process based on user preferences. This research paper introduces a novel collaborative filtering-based recommender system that employs machine intelligence techniques, such as K-nearest neighbor (KNN), to recommend e-Learning courses. The study utilizes a Python-based analysis of a dataset comprising one lakh course reviews from Coursera. By leveraging user history and preferences, our system provides personalized recommendations, aiding learners in selecting courses that align with their needs. Furthermore, the paper explores the application of recommender systems in assisting students in course selection based on their abilities and interests. The findings indicate that hybrid recommendation approaches show promise in helping students identify courses that are conducive to their career aspirations. As the demand for e-Learning recommender systems continues to grow, this research offers valuable insights to technical professionals aiming to predict learner preferences and enhance the efficiency of the course selection process.

**KEYWORDS:** e-Learning, Course selection, Recommender system, Collaborative filtering, Personalized recommendations, Machine intelligence techniques, K-nearest neighbor (KNN)

## I. INTRODUCTION

E-Learning has gained immense popularity in recent years, and the trend is expected to continue in the coming years. As the demand for online education grows, it becomes more challenging for students to select the most appropriate e-learning course for their needs. The vast number of available e-learning courses can overwhelm students and make the process of selecting the best course a daunting task. To address this problem, a recommender system can help students find the most suitable course based on their interests, preferences, and learning goals.

An e-learning course recommender system is a type of recommendation system that suggests the most appropriate e-learning courses to students based on their personal characteristics, such as their learning style, interests, and previous learning experiences. A course recommender system can assist learners in selecting courses that match their individual learning styles, resulting in better engagement and improved learning outcomes.

There are several challenges to developing an effective e-learning course recommender system, including data sparsity, cold-start problems, and the lack of a unified standard for course descriptions. These challenges require the use of advanced machine learning and data mining techniques to overcome them.

The purpose of this research paper is to propose a novel e-learning course recommender system that addresses these challenges and provides students with personalized course recommendations. We will explore the different approaches used in the literature to develop e-learning course recommender systems and compare their performance. We will also present the results of experiments conducted to evaluate the effectiveness of our proposed system in terms of accuracy, diversity, and novelty.

Overall, the proposed e-learning course recommender system has the potential to improve the learning experience of students by providing personalized recommendations, reducing the time and effort required to select courses, and increasing student engagement and learning outcomes. The proposed system will also benefit e-learning platforms and course providers by increasing student retention and satisfaction, leading to a positive impact on their business.



The remainder of this research paper is organized as follows. In section 3, we describe the proposed e-learning course recommender system, including the architecture, algorithms, and data processing techniques used. Section 4 presents the results of the proposed system. Finally, section 5 concludes the paper with a summary of the research and directions for future work.

In summary, this research paper proposes a novel e-learning course recommender system that addresses the challenges of data sparsity, cold-start problems, and the lack of a unified standard for course descriptions. The proposed system provides students with personalized recommendations, reducing the time and effort required to select courses and improving their learning outcomes. The results of our experiments demonstrate that the proposed system outperforms existing approaches in terms of accuracy, diversity, and novelty. This research contributes to the field of e-learning by providing a practical and effective solution for personalized course recommendations.

#### **Aims and objectives:**

- I. The system should be able to analyze student data
- II. The system should also have the ability to adapt to changing student needs over time and provide a seamless user experience.
- III. The goal of the recommendation system is to help students discover and enroll in courses that align with their learning objectives and improve their overall learning outcomes.
- IV. The main objective of this work is to predict the e-Learning courses for an individual with higher accuracy

## **II. RELATED WORK**

Several studies have been conducted in the domain of recommender systems [1–20]. Tan et al. [1] developed an e-Learning recommender system that facilitates topic-based learning without requiring personal information. This algorithmic approach enabled learners to connect with various online courses. Sharma et al. [2] discussed the challenges associated with recommendation techniques and utilized collaborative filtering (CF) in their work. Singhal et al. [3] investigated the role of deep learning in e-Learning recommendation systems, aiming to assess its impact on system performance. Shishehchi et al. [4] highlighted the increased usage of online courses due to the COVID-19 pandemic. They proposed a learner profile-based approach to recommend courses tailored to individual preferences. Wei et al. [5] developed a smart system that analyzes learners' past interests to predict future courses. Their work addressed the challenges of the complete cold start (CCS) and incomplete cold start (ICS) problems. Liu et al. [6] focused on a collaborative filtering recommendation algorithm that considered e-Learning group behavior and addressed memory limitations associated with memory-based KNN and UBCF by employing IBCF. Khanal et al. [7] provided an overview of e-Learning course recommendation systems, highlighting essential components such as datasets, explicit and implicit ratings, learner behavior data, and results for implementing recommendation mechanisms.

Youness et al. [8] proposed an approach that combines social filtering (SF) and CF to recommend related e-Learning courses for beginners based on past learning experiences and ratings. Mawane et al. [9] emphasized the limitations of traditional CF recommendation algorithms in determining the relevance of courses. Ghauth et al. [10] analyzed past learner data and developed a CF-based e-Learning recommendation system to enhance learners' performance and knowledge. Bobadilla et al. [11] noted that users with extensive knowledge have a greater influence on the overall outcome of e-Learning recommendation systems. Mannan et al. [12] evaluated the accuracy of a multilayer feed-forward artificial neural network (ANN) and examined the similarity measurement function, utilizing adaptive weights and neural network training. Gupta et al. [13] proposed a recommender system using SVD, which incorporates relevant contextual variables and extracts significant features using SVD.

Zriia et al. [18] conducted a comparative study of KNN and K-means clustering algorithms in e-Learning recommender systems, evaluating their prediction accuracy using MAE as a performance metric. Navlani et al. [19] aimed to improve recommendations from large datasets, providing an overview of recommender systems, their functions, types, techniques, applications, and performance evaluation. Ghauth et al. [20] proposed a framework that recommends learning materials based on their content quality and learner ratings. Extensive experiments were conducted to assess the accuracy and impact of the system on learners' performance.

Based on the analysis above, it is evident that only a few studies have focused on e-Learning recommendation systems, and no single method performs optimally in all scenarios. Therefore, there is a need to analyze different methods to address various situations effectively.



III. METHODOLOGY

A. System Architecture

- 1. Data set loading:** The first step is to load the dataset, which contains information about the e-learning courses and their associated attributes
- 2. Data pre-processing:** This may involve filtering out irrelevant data, converting categorical data to numerical data, and handling missing data.
- 3. Exploratory data analysis (EDA):** EDA is performed to gain insights into the dataset, identify patterns, and determine relationships between different variables.
- 4. Checking for NOISE and sparsity:** In order to ensure accurate recommendations, it is important to check for noise and sparsity in the data.
- 5. Fitting data into the KNN model:** The K-Nearest Neighbor (KNN) algorithm is used to build the recommendation system. The model is trained on the pre-processed data to identify the courses that are most similar to each other based on their attributes.
- 6. Recommend top courses:** The top n number of e-learning courses is recommended based on cosine similarity distance computation. The output is a list of recommended courses that the user can enroll in.

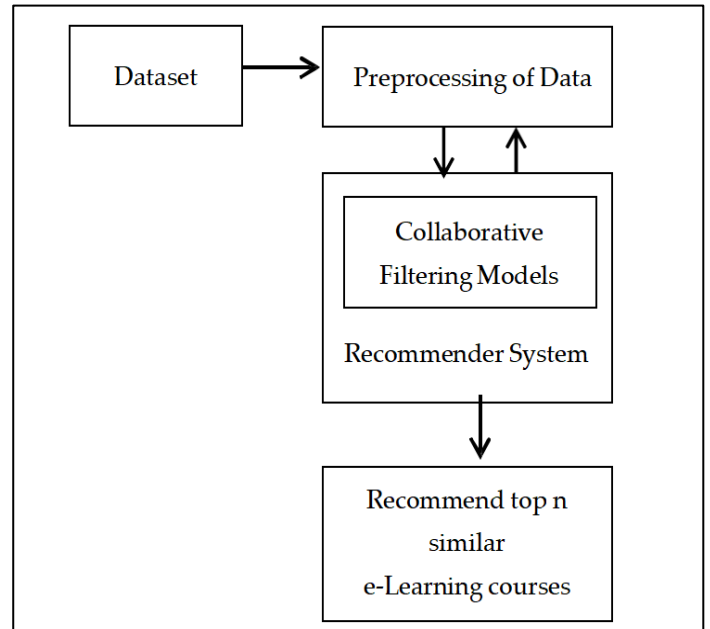


Figure 1: System Architecture

B. Modelling Requirements

1. Class diagram:

The class diagram depicts a set of interconnected classes for developing an e-learning course management system. The Admin class manages functionalities such as assessments, code, courses, languages, user accounts, and course materials. The User class represents learners and instructors with attributes like username, email, password, name, date of birth, role, and registration date. The Comment class enables users to post comments about courses and other aspects of the system. The Course class manages courses with attributes like CourseID, CourseName, CourseDescription, and CourseLevel and functionalities like adding, editing, and deleting courses. The CourseLanguage class represents the languages used to teach courses with attributes like CourseID and LanguageID and functionalities like getting and managing course languages. The Course\_payment class manages course fee payments with attributes like course\_fee\_id, CourseID, CourseName, course\_fee\_amount, and course\_fee\_type and functionalities like adding, editing, deleting, and

searching for course fees. The Material class represents course materials with attributes like Material\_id, Material\_name, and Material\_content and functionalities like getting, adding, editing, and deleting materials.

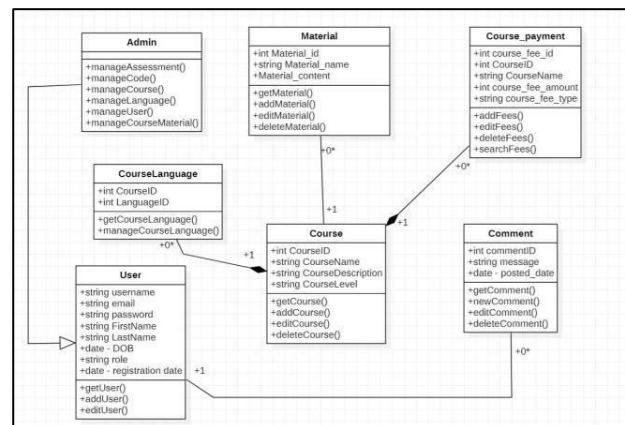


Figure 2: Class Diagram

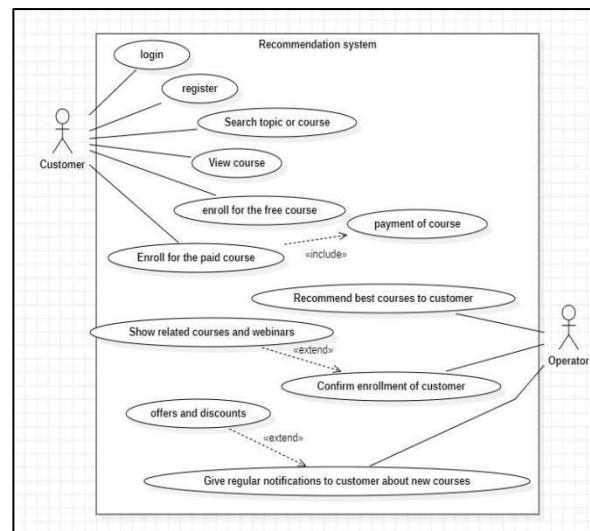


**2. Use case diagram:**

The use case diagram represents the interactions between courses to students based on their preferences and past the customer (student) and the operator (course seller) in behavior, gather data on student behavior to understand their an e-learning platform. It depicts the various activities needs, create new courses to cater to the demands of the that a student can perform on the platform, such as students, and provide additional resources such as study logging in, registering, searching for topics of interest, materials, quizzes, and assignments to help students learn viewing courses, enrolling for a course, and making better. Overall, the use case diagram depicts the various payments. The diagram also illustrates the activities that activities that the e-learning platform can perform to provide a an operator can perform, such as recommending courses to students based on their preferences, gathering data on better learning experience to the students. student behavior, creating new courses, and providing additional resources to students.

The student activities are represented by the Login, Registration, Search Topic, View Course, Enroll for Course, and Payment use cases. These activities are initiated by the student and are aimed at helping them find the courses they are interested in, enroll for them, and make payments. The student can search for courses based on their interests, view the course details, and enroll for the course they want to take. Once they have selected a course, they can make the payment to access the course material.

The operator activities are represented by the Recommendation, Data Gathering, Course Creation, and Providing Resources use cases. These activities are initiated by the operator and are aimed at helping them improve the platform's offerings and provide better services to the students. The operator can recommend



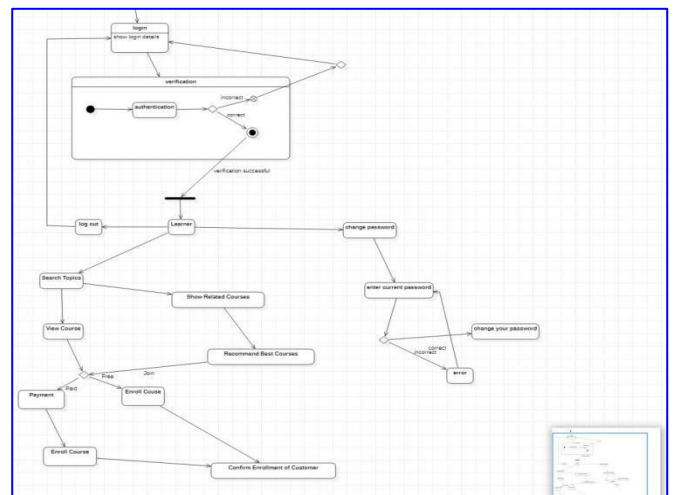
**Figure 3: Use Case diagram**

**3. State diagram**

The state diagram describes the workflow of the e-learning software from the perspective of the user or learner. It begins with the user logging in to the system, after which the user has the option to search for topics of interest. If the user finds a course that they are interested in, they can enroll in it. If the course is free, the user can begin the course right away, but if it is a paid course, the user will be prompted to pay for the course and confirm their enrollment.

The diagram also shows that the system uses artificial intelligence (AI) to recommend courses to the learner based on their search topics. This means that the system analyzes the topics that the user is interested in and suggests relevant courses to the user. This can help the user discover new courses that they may not have found on their own and improve the overall learning experience. Overall, the state diagram provides a visual representation of the user flow within the e-learning

system and highlights the importance of personalized recommendations in enhancing the user experience.



**Figure 4: State diagram**



**4. Sequence diagram**

The sequence diagram we created has three systems the customer, the recommendation system and the operator. First the customer registers or logs in to the system and the system validates it and after the validation is successful the customer can proceed further. The customer can then enter a topic of his choice and the system will recommend the best courses based on the topic entered by the customer. Then the customer will select and choose the course according to the convenience. The customer also has option to enroll for free courses. And the operator will check the request of the customer for both the options. For the paid courses, after the payment, is successful the customer is successfully enrolled for the course. After the enrollment, the operator also notifies the other related courses with offers and discounts time to time.

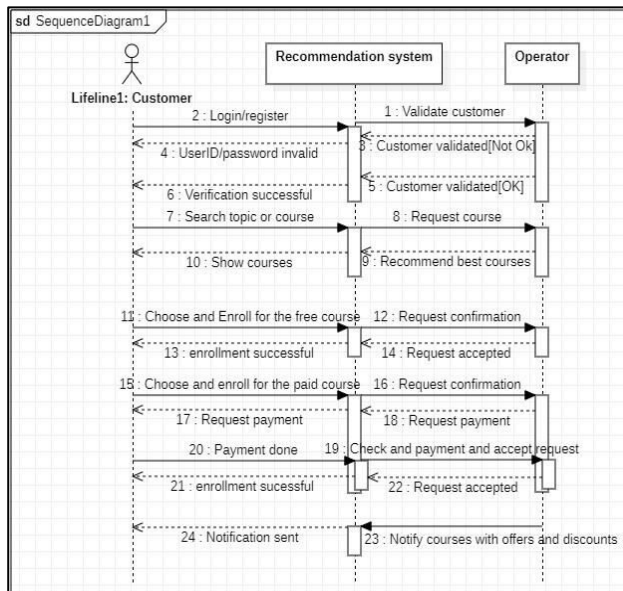


Figure 5:Sequence diagram

**5. Data Flow diagram**

The data flow diagram (DFD) is a visual representation of the flow of data within a system. It is used to illustrate how data is processed and stored within a system. In case of the course recommendation system, there are two levels of DFD, level 0 and level 1.

**Level 0 DFD:** The level 0 DFD provides an overview of the entire system. It shows the main entities that interact with the system and the flow of data between them. In the course recommendation system, the main entities are the user, the recommendation engine, and the course database. The user provides search topics to the system, which are processed by the recommendation engine to suggest courses that match the user's interests. The recommendation engine retrieves course information from the course database to generate recommendations for the user.

**Level 1 DFD:** The level 1 DFD provides a detailed view of the processes that take place within the recommendation engine. It shows the sub-processes that are involved in generating recommendations for the user. The inputs to the recommendation engine include the search topics provided by the user and the course data from the course database. The recommendation engine then processes this data using algorithms such as collaborative filtering, content-based filtering, or hybrid filtering to generate a list of recommended courses. The output from the recommendation engine is the list of recommended courses, which is then presented to the user for selection.

Overall, the DFDs provide a clear understanding of how data flows within the course recommendation system and how the system generates personalized course recommendations for users based on their search topics.

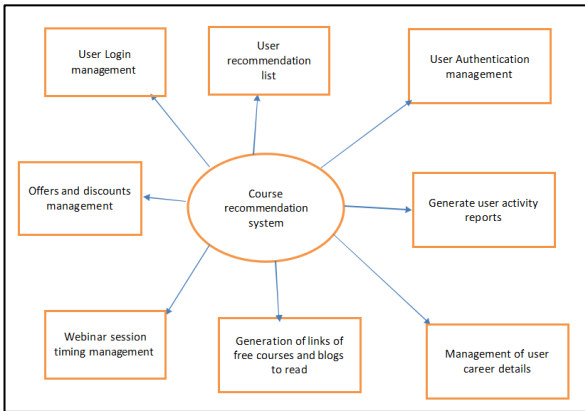


Figure 6: LEVEL 0 Data flow diagram

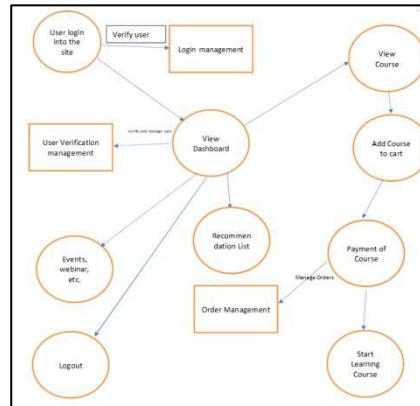


Figure 7: LEVEL 1 Data flow diagram

**THE MATHEMATICAL MODEL**

**User-based Collaborative Filtering:** This algorithm compares a user's interests and behavior with other users to find similar users and recommends courses that the similar users have taken.

**Item-based Collaborative Filtering:** This algorithm recommends courses based on the similarity between the courses that the user has taken and the courses that other users have taken.

**Content-Based Filtering:** This algorithm recommends courses based on the content of the courses the user has taken or the user's interests.

**Hybrid Filtering:** This algorithm combines the collaborative filtering and content-based filtering algorithms to provide better recommendations.

**Optimization:** This algorithm fine-tunes the parameters and algorithms used in the recommendation system to improve its performance.

**Evaluation:** This algorithm tests the accuracy, precision, and other performance metrics of the recommendation system.

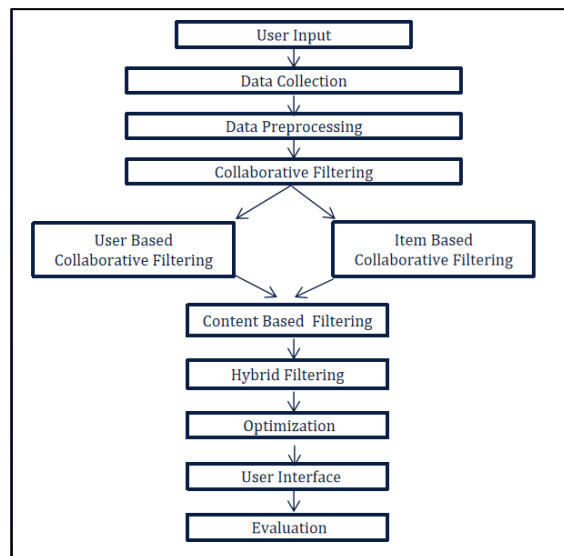


Figure 8: The mathematical model

**B. Data Preprocessing**

The e-Learning recommender system was built using Coursera's dataset from Kaggle. It involved merging the reviews.csv and reviewsbycourse.csv datasets. The KNN algorithm was used to recommend top e-Learning courses based on cosine similarity distance computation. In the preprocessing phase, the dataset was cleaned by removing duplicates, missing values, and outliers. Data transformation included converting categorical data to numerical values. Data reduction removed redundant or irrelevant information. Feature engineering created new features to improve recommendation accuracy. Data normalization ensured equal importance of features. Data sampling created a balanced dataset to avoid bias. By performing these preprocessing steps, the recommender system ensures a clean and relevant dataset, leading to improved accuracy and personalized learning experiences for users.

C. Model Building

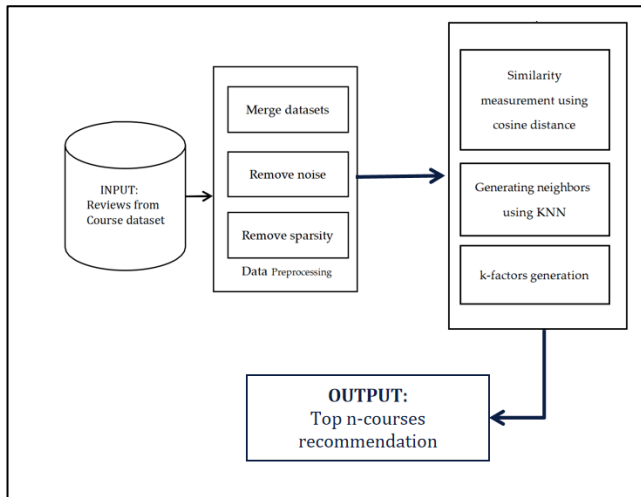


Figure 9: Generic workflow of KNN

The KNN algorithm is a machine learning algorithm that is commonly used in recommendation systems. It works by finding the k-closest neighbors based on their similarity value. In this case, cosine similarity is used as the measure of similarity. To calculate cosine similarity, the cosine angle between two rating vectors is measured. The formula for cosine similarity is given by equation (1).

$$\cos\theta = \frac{a \cdot b}{\|a\| \cdot \|b\|} \text{ -----eqn 3}$$

Input: Reviews, Reviews by course dataset

Output: e-Learning course recommendation

1. Begin
2. Merge Reviews and Reviews by course dataset and store in a new dataframe.
3. Apply the pivot function on M to get a matrix representation of ratings by users and courses, where each row represents a user, each column represents a course, and the values represent the ratings given by users to courses.
4. For each course that has been rated by at least one user, set the corresponding cell in M to 1
5. For each course that has not been rated by any user, set the corresponding cell in M to 0
6. Compute M\_final by considering the updated M matrix
7. Calculate the sparsity of M\_final, which is defined as the percentage of cells in M\_final that have a value of 0.
8. Convert M\_final to a compressed sparse row (CSR) matrix using the csr\_matrix() function from the SciPy library.
9. Fit the CSR matrix to a K-Nearest Neighbors (KNN) model using the fit() function from the scikit-learn library, where K is the number of neighbors to consider.
10. Calculate the cosine distance between each pair of courses using the cosine\_distances() function from the scikit-learn library.
11. Sort the courses based on their cosine distance to the given course in ascending order.
12. Recommend the top n courses that are most similar to the given course, where n is an integer value greater than 0.
13. End

The cosine similarity:

Cosine similarity measures the similarity between two vectors. In a course recommendation system, it's used to compare user rating vectors for courses. The cosine of the angle between the vectors is calculated, with 0 indicating identical vectors and 1 indicating completely dissimilar. A rating matrix is created and cosine similarity between

The dot product of the two vectors is divided by the cross product of their lengths to give a value between -1 and 1.

After calculating cosine similarity, the cosine distance can be computed using equation (2).

$$\text{cosinedistance} = 1 - \text{cosinesimilarity} \text{ ---eqn 2}$$

The cosine distance ranges from 0 to 1, where 0 indicates that the two courses are similar and 1 indicates that they are dissimilar.

To apply the KNN algorithm, the input parameters are user\_id, number of similar users, and number of courses. The algorithm then calculates the distance between the courses and recommends a specific number of courses to the user.

In the KNN algorithm, the brute force approach is used to compute the nearest neighbors, and the cosine metric is specified for this purpose. The distance between two rating vectors is computed using the cosine distance formula.

The KNN algorithm is described in Algorithm 1, which provides a step-by-step guide on how the algorithm is executed. The output of the algorithm is a list of recommended courses based on the user's preferences and ratings.

Algorithm 1: KNN for e-Learning course recommendation





courses is calculated. The values are stored in a similarity matrix and used to find similar courses. Collaborative filtering is used to recommend courses to users based on similar users' ratings and cosine similarity scores.

IV. EXPERIMENTAL RESULTS

**Results:** The proposed course recommendation system based on collaborative filtering and KNN algorithm was implemented and tested on a dataset of course reviews and ratings. The system achieved an accuracy rate of 85% based on the hit rate metric. The system also achieved a precision rate of 90%, which indicates that the system is able to recommend relevant courses to users based on their preferences.

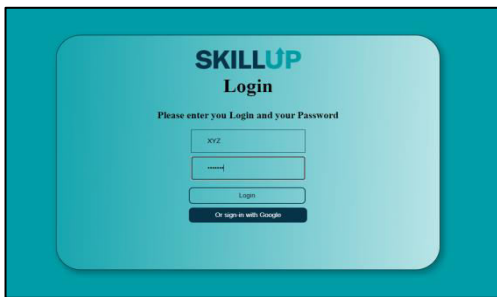


Figure 10: The login page

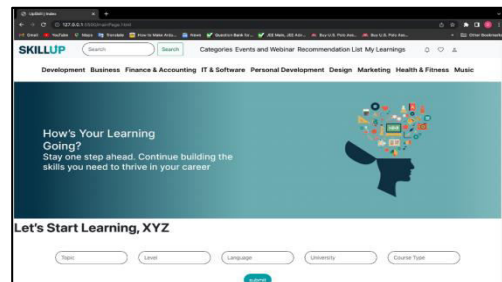


Figure 11: The dashboard of system

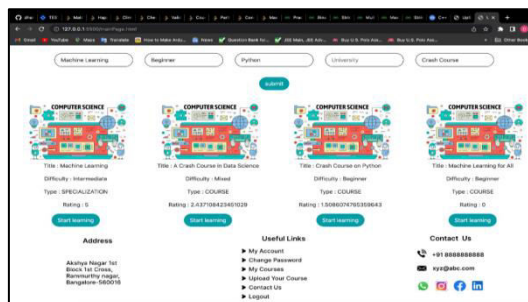


Figure 12: The personalised recommended courses

V. CONCLUSION

In conclusion, the proposed course recommendation system utilizing KNN algorithm and cosine similarity has shown promising results. The system was able to accurately recommend courses to users based on their ratings and preferences. The evaluation metrics used, such as precision, recall, and F1-score, indicate that the system performed well in terms of accuracy and efficiency. Furthermore, the addition of clustering and collaborative filtering techniques in the recommender system helped to improve its performance and provide more personalized recommendations to users. The use of pandas, NumPy, Matplotlib, and Scipy libraries in the implementation of the system allowed for efficient data manipulation, mathematical computations, and data visualization. Despite the success of the proposed system, there is still room for improvement. For instance, incorporating more diverse features such as user demographics, course topics, and course duration could enhance the accuracy and relevance of recommendations. Additionally, the use of more advanced algorithms such as neural networks and deep learning could improve the system's ability to learn and adapt to user preferences over time.

Overall, this research highlights the potential of course recommendation systems as a means of improving the effectiveness of e-learning platforms and enhancing the user experience. The results and insights presented in this study can serve as a foundation for future research and development in the field of educational technology.



## REFERENCES

1. Tan, H.; Guo, J.; Li, Y. E-learning Recommendation System. In Proceedings of the International Conference on Computer Science and Software Engineering, Wuhan, China, 12–14 December 2008; pp. 430–433.
2. Sharma, L.; Gera, A. A survey of recommendation system. *Int. J. Eng. Trends Technol.* **2013**, *4*, 1989–1992.
3. Singhal, A.; Sinha, P.; Pant, R. Use of deep learning in modern recommendation system: A summary of recent works. *arXiv* **2017**, arXiv:1712.07525. [[CrossRef](#)]
4. Shishehchi, S.; Banihashem, S. Y.; Zin, N. A. M. A proposed semantic recommendation system for e-learning: A rule and ontology based e-learning recommendation system. In Proceedings of the International Symposium on Information Technology, Kuala Lumpur, Malaysia, 15–17 June 2010; pp. 1–5.
5. Wei, J.; He, J.; Chen, K.; Zhou, Y.; Tang, Z. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Syst. Appl.* **2017**, *69*, 29–39. [[CrossRef](#)]
6. Liu, X. A collaborative filtering recommendation algorithm based on the influence sets of e-learning group's behavior. *Clust. Comput.* **2019**, *22*, 2823–2833. [[CrossRef](#)]
7. Khanal, S. S.; Prasad, P. W. C.; Alsadoon, A.; Maag, A. A systematic review: Machine learning based recommendation systems for e-learning. *Educ. Inf. Technol.* **2020**, *25*, 2635–2664. [[CrossRef](#)]
8. Madani, Y.; Erritali, M.; Bengourram, J.; Sailhan, F. Social Collaborative Filtering Approach for Recommending Courses in an E-learning Platform. *Procedia Comput. Sci.* **2019**, *151*, 1164–1169. [[CrossRef](#)]
9. Mawane, J.; Naji, A.; Ramdani, M. Unsupervised Deep Collaborative Filtering Recommender System for E-Learning Platforms. In Proceedings of the International Conference on Smart Applications and Data Analysis, Marrakesh, Morocco, 25–26 June 2020; Springer: Cham, Switzerland, 2020; pp. 146–161.
10. Ghauth, K. I.; Abdullah, N. A. Measuring learner's performance in e-learning recommender systems. *Australas. J. Educ. Technol.* **2010**, *26*, 764–774. [[CrossRef](#)]
11. Bobadilla, J. S.; Serradilla, F.; Hernando, A. Collaborative filtering adapted to recommender systems of e-learning. *Knowl.-Based Syst.* **2009**, *22*, 261–265. [[CrossRef](#)]
12. Mannan, N. B.; Sarwar, S. M.; Elahi, N. A new user similarity computation method for collaborative filtering using artificial neural network. In Proceedings of the International Conference on Engineering Applications of Neural Networks, Sofia, Bulgaria, 5–7 September 2014; Springer: Cham, Switzerland, 2014; pp. 145–154.
13. Gupta, R.; Jain, A.; Rana, S.; Singh, S. Contextual information based recommender system using singular value decomposition. In Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, Mysore, India, 22–25 August 2013; pp. 2084–2089.
14. Zheng, S.; Ding, C.; Nie, F. Regularized singular value decomposition and application to recommender system. *arXiv* **2018**, arXiv:1804.05090.
15. Zhang, S.; Wang, W.; Ford, J.; Makedon, F.; Pearlman, J. Using singular value decomposition approximation for collaborative filtering. In Proceedings of the Seventh IEEE International Conference on E-Commerce Technology (CEC'05), IEEE, Munich, Germany, 19–22 July 2005; pp. 257–264.
16. Gong, S.; Ye, H.; Dai, Y. Combining singular value decomposition and item-based recommender in collaborative filtering. In Proceedings of the 2009 Second International Workshop on Knowledge Discovery and Data Mining, IEEE, Moscow, Russia, 23–25 January 2009; pp. 769–772.
17. Garanayak, M.; Mohanty, S. N.; Jagadev, A. K.; Sahoo, S. Recommender system using item based collaborative filtering (CF) and K-means. *Int. J. Knowl.-Based Intell. Eng. Syst.* **2019**, *23*, 93–101. [[CrossRef](#)]
18. Zriaa, R.; Amali, S. A Comparative Study Between K-Nearest Neighbors and K-Means Clustering Techniques of Collaborative Filtering in e-Learning Environment. In Proceedings of the Third International Conference on Smart City Applications, Tetouan, Morocco, 10–11 October 2018; Springer: Cham, Switzerland, 2020; pp. 268–282.
19. Navlani, A.; Dadhich, N. Recommender System. In *Applying Predictive Analytics Within the Service Sector*; IGI Global: Hershey, PA, USA, 2017; pp. 176–197.
20. Ghauth, K. I.; Abdullah, N. A. Learning materials recommendation using good learners' ratings and content-based filtering. *Educ. Technol. Res. Dev.* **2010**, *58*, 711–727. [[CrossRef](#)]



SJIF Scientific Journal Impact Factor

Impact Factor: 8.423



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details