

Smart Course Recommendation System

P. Shree Varshini

Department of Computer Science
and Engineering
Vardhaman College of Engineering
Hyderabad-Telangana, India

L. Abhivardhan Reddy

Department of Computer Science
and Engineering
Vardhaman College of Engineering
Hyderabad-Telangana, India

M. Surya Teja

Department of Computer Science
and Engineering
Vardhaman College of Engineering
Hyderabad-Telangana, India

Dr S. Srinivas Reddy

Department of Computer Science
and Engineering
Vardhaman College of Engineering
Hyderabad-Telangana, India

Dr. Tamilarasi M

Department of Mathematics
Sri Eshwar College
of Engineering Coimbatore,
Tamil Nadu, India

Abburri Ramesh

Department of Computer Science
and Engineering
Narasaraopeta Engineering College
Palnadu dist, A.P.,India

Abstract—Currently due to the digital learning environment, students are struggling to choose the most suitable courses that match their interests, skills, and career goals. As there are numerous courses appears online, students often get confuse on what course should be taken in-order to upskill their knowledge. Students often struggle to choose the right course because they have many options but lack guidance in selecting one that matches their skills and areas of interest. The Smart Course Recommendation System aims to address this challenge by providing personalized course suggestions using Machine Learning (ML) techniques. The system looks at different factors such as a student's academic background, performance, interests, and learning preferences to generate accurate and correct recommendations. By using data-driven insights helps students make smart choices, stay intrested, and promotes continuous learning. The project aims to connect student's goals with their course selection, which will lead to better learning and overall satisfaction of students. The methodologies that are used for filtering the courses based on their interests and skills are Content-based filtering, Skill similarity matching algorithm - it ensures that the recommendations are relevant to the student's preferences and compares both skills and calculates number of matching skills. Top-N ranking mechanism (KNN) approach - based on their previous scores the system provides top 5 courses and shows them to the user. Rule based conversational AI Assistant that helps students explore more about their interests. The proposed solutions aims to assist students in making informed decisions about their learning path and career development. Experimental results shows that the system can effectively give relevant and personalized course suggestions, which makes the learning experience better for users.

Keywords: Machine Learning, E-Learning, Personalization, Student Performance, Data-Driven Decision Making, Content-Based Filtering, Skill Matching Algorithm, Personalized Learning, Top-N Recommendation, Career Guidance System.

I. INTRODUCTION

In recent years, due to the rapid increase in technology students often gets confused on what should be done next? which field should be chosen? what tasks should be implemented

further in order to get into that job? Due to the lack of perfect guidance, they are choosing the career growth which is in trend irrespective of the skills and interests they acquire. Many career options in software such as Data Science, Artificial Intelligence, Web Development, Cyber Security, and many more are in demand but individuals are unaware of which one should be chosen. To avoid the confusion and to choose the right path Smart Course Recommendation is implemented. In the past, traditional methods like searching every topic, taking opinions from others and starting that course then after they don't want to continue in the same course. This is such a waste of time and no knowledge was also gained from it as it always feels like a burden rather than being excited or motivated. As selecting a right course is crucial to a student for their professional future they should always focus on the correct career goal and acquire skills regarding that course instead of searching through all the available options. Selecting a perfect one based on their interests and the skills that they already have helps them in motivating themselves to be in the same path and also helps in continuous learning which leads in gaining full knowledge in that course that helps in achieving a good professional career in that field. In the field of education, these recommendation system assists students by suggesting relevant courses that matches their interests, skills, learning objectives, and knowledge level. This research paper proposes a Smart Course Recommendation System which mainly focuses on providing the perfect recommendation of a particular course based on the individual interests and skills. This helps them in finding the correct course they should be acquired for future purpose and they will get to know how much percentage of knowledge they will be gained by completing that particular course. This recommender system uses a content-based filtering approach to analyze the relation between user inputs and course attributes which are embedded in a dataset. A skill matching mechanism is used to compare the user's skill with the skills required for different courses.

From these comparisons, a similarity score is calculated and the most relevant top N courses will be shown. Based on that course the student has to learn, acquire skills and ensure continuous learning. It is a web-based architecture that contains user-friendly interface for input collection and backend system for processing generating recommendations. Python in Jupyter Notebook is the language used for backend purpose whereas the course dataset contains course name, required skills, interest domain, and price. The main objective of this is to help student overcome the difficulty of selecting appropriate courses by providing intelligent and personalized recommendations. To apply the business approach to improve the Smart Course Recommendation System, the work expands the model through adding SHAP (SHapley Additive exPlanations) to explainable analytics. Although the base system suggests that courses be selected with the aid of content based filtering and similarity scoring, the introduction of SHAP allows the interpretation of the model predictions especially in predicting course success or profitability. Through the examination of the input of the major features like the price, subscribers, reviews, and time of the course, the system gives a clear information on what makes some of the courses to be recommended. This enhances transparency and user confidence, as well as business decision making; as it is able to reach factors that contribute to course performance and, therefore, make recommendations consistent with the needs of the learners and profitability of the platform.

II. LITERATURE SURVEY

Many researchers have worked on recommender systems and educational recommendation systems. In the recent times, these online recommendation systems plays a significant role as it helps in providing a close course recommendation based on the interests, skills, and knowledge level that helps in reducing time in searching all the available courses. Hence, the growth of these systems increases used for e-commerce, online suggestion courses, etc. It is a need for the students who doesn't have any idea on how they should be proceeded further i.e, in what categories of courses, and on their professional career. In the previous years, researchers used several techniques such as collaborative filtering, content based filtering, Hybrid recommendation systems , Education data mining, Machine-Learning based filtration . In Collaborative filtering, it identifies the patterns of what the user has been given and the data sets that are embedded in the backend. By identifying the patterns it recommends the courses which are closely similar to user skills and interests. Common methods like matrix factorization (Single value decomposition), Nearest Neighbours, and Neural Collaborative Filtering are used mostly. These are applied in the applications like movie recommendation, music recommendation example: Netflix, Spotify, E-commerce product recommendations like Amazon, and Content recommendation on social media and news sites. In Content based-filtering it is a recommendation engine technique that suggests items similar to those a user liked in the past by analysing item features. Metadata like genre, keywords or description. It

builds user profile based on the individual attributes, interests. In Hybrid recommendation systems, it is a combination of content-based filtering and collaborative filtering methods. These are mainly used for obtaining accurate results and also increases the number of products recommended. It helps in scalability and gets ordered lists of results and also manages dynamic scenarios [12]. Another method called Educational Data Mining (EDM) is an interdisciplinary field that applies data mining, machine learning and also statistics to data from educational settings like intelligent tutoring systems, LMS, etc. This is used to analyse student performance, understanding learning process, and improve educational outcomes. In this the data is collected from e-learning platforms and administrative records. Common methods like clustering, regression analysis, classification were also used for effective results. In collaborative filtration, they calculated users similarity using cosine similarities metrics and Pearson correlation coefficient (PCC) [13].

$$sim_{uv} = \cos(\vec{r}_u, \vec{r}_v) = \frac{\vec{r}_u \cdot \vec{r}_v}{\|\vec{r}_u\|_2 \times \|\vec{r}_v\|_2} = \frac{\sum_{i \in I_{uv}} \vec{r}_{ui} \cdot \vec{r}_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2} \sqrt{\sum_{i \in I_v} r_{vi}^2}}$$

$$sim_{uv} = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}}$$

whereas $\bar{r}_u$ and $\bar{r}_v$ represent the average ratings from u and v .

After the filtration techniques they used Next Nearest approach in order to obtain the similar patterned skills. Then ratings were predicted using the formula:

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N_u} sim_{uv}(r_{vi} - \bar{r}_v)}{\sum_{v \in N_u} |sim_{uv}|}$$

Where N_u shows the collection of users who are same as user u [13]. Another research based on recommender systems they introduced Monotonic non linear State-Space(MNNS).This tracks how a user's skill grow over time. It represents skills as binary latent states that increases when the student gain experience. It used data from LinkedIn and provides choices based on it. The MNNS model also enables skill gap identification – it helps in finding what skills the user needs to reach a target career goal. Career path recommendation- they used this to suggest feasible job transitions to achieve that goal. Reconstruction of skill acquisition throughout a person career[14].

$$p_{\theta}(\mathbf{x}_{1:T}, \mathbf{z}_{1:T} | \mathbf{u}_{1:T}) = p_{\theta_z}(\mathbf{z}_1) \prod_{t=2}^T p_{\theta_z}(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{u}_t) \prod_{t=1}^T p_{\theta_x}(\mathbf{x}_t | \mathbf{z}_t),$$

where $p_{\theta_z}(\cdot)$ is the latent skill state transition model and $p_{\theta_x}(\cdot)$ is the career experience emission model; θ_z and θ_x denote the set of parameters in these models, respectively, and $\theta = \{\theta_x, \theta_z\}$ denotes the set of all model parameters.

Here, it also provides interpretable and actionable recommendations[14]. However, the system has limitations because it relies on historical career data and does not consider rejected job offers or real-life constraints. There are few limitations from the past researches that are it as it requires large datasets, it gets high computational complexity and that results in delayed outcomes. It also lacks personalization and limited real-time recommendation. Several studies are conducted in recommendation systems for online learning, many existing solutions have certain limitations that reduce their effectiveness in guiding students towards appropriate courses. This recommender system uses a content-based filtering approach to analyze the relation between user inputs and course attributes which are embedded in a dataset. A skill matching mechanism is used to compare the user's skill with the skills required for different courses. From these comparisons, a similarity score is calculated and the most relevant top N courses will be shown. Based on that course the student has to learn, acquire skills and ensure continuous learning. It is a web-based architecture that contains user-friendly interface for input collection and backend system for processing generating recommendations. Python in Jupyter Notebook is the language used for backend purpose whereas the course dataset contains course name, required skills, interest domain, and price. The main objective of this is to help student overcome the difficulty of selecting appropriate courses by providing intelligent and personalized recommendations.

Also, new developments point towards the need to have model interpretability. Most of the traditional recommendation systems are black-box models, and it therefore proves difficult to explain how the recommendations are made. To solve this, explainable AI methods like SHAP (SHapley Additive exPlanations) were proposed. SHAP gives the user and the developer a sense of the contribution made by features and how it affects the predictions. Nonetheless, the existing systems concentrate on the accuracy or interpretability of recommendations, but not common systems combine the two in a way that works. This leaves a loophole in the formation of systems that are precise and justifiable.

SHAP (SHapley Additive exPlanations) are introduced into the system to improve the level of transparency and interpretability. SHAP analysis tells how much any given feature (price, reviews, and duration) contributes to the model predictions so that the users and interested parties can know why a given course is recommended.

III. RESEARCH METHODOLOGY

This design is a system that helps people learn things. This is more like a guide that gives people choices and shows them

what can happen when they make those choices. The system is very useful for people who want to do their jobs. It looks the jobs that're available and figures out what people need to know to do those jobs. The system uses computer programs that can understand what people are saying to find the information.

A. Work Flow: The workflow illustrates the working process of the proposed Smart Course Recommendation System. Firstly, the system collects user interests and skills as input. The input is then processed to prepare it for analysis. Content-based filtering is applied to identify courses after preprocessing which is related to the user's interest from the course dataset. Thereafter, a skill matching algorithm compares the user's skills with the skills required for each course and calculates a similarity score. Based on this score which is opted, the system ranks the courses and selects the top N courses. Finally, the system generates and displays the recommended course list to the user. This methodology is a combination of Content-Based Filtering, Skill matching algorithms, and K-NN approach(Top N course list)

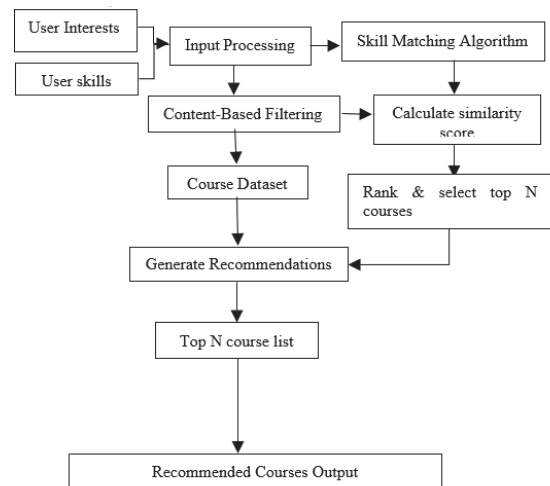


Fig. 1. Flow chart of the proposed approach

After calculating the similarity scores, the system uses a ranking mechanism to list the courses based on how closely they are related to the user's interests and skills. The most relevant courses are then selected using a Top-N recommendation approach, where the system provides a list of the best matching courses to the user accordingly.

The proposed system is implemented as a web-based application using a client-server architecture. The front-end interface consists of interests and skills which the user has to enter them, while the backend server processes the input and generates recommendations based on the inputs. The Flask framework is used to build the back-end, and helps in processing the data which are used to manage the course dataset. RESTful APIs are used to send messages between the front end and the back end.

Overall, the system helps to simplify the decision-making process for students by providing intelligent and personalized course recommendations. The analysis of user inputs and

course data, as interpreted by the system, assists the students in pointing out those courses which are most appropriate to them according to the learning path and career building. It is a dynamic architecture of the system that facilitates the ease with which new courses and category of skills may be added to it in the future.

B. Data Collection: This recommendation system is based on the CSV(Comma Separated Values) file to store its dataset. The file includes a lot of information regarding a course. These features involves course name, area of interest, skills within the course, and its level of difficulty. Each course in the dataset is associated with the specific field, such as artificial intelligence, web development, data science, or others. The required skills attribute identifies the required skills of what a person must have to complete or comprehend the course successfully.

The data that is gathered is arranged in a systematic form in such a way it can be processed easily by the recommendation system. The data set is represented as rows where each course is represented as a row and each attribute of the course as a column. Such a systematized representation simplifies the process of searching, filtering and analyzing course information during the process of recommendations.

C. Data Preprocessing: Preprocessing of data is carried out to obtain the dataset and user input prepared to give the correct course suggestions. Data cleaning is performed first to eliminate any missing or inconsistent values of the data set. Then, all written information. similarity in interests and talent is transformed into small letters to. do not have too many changes. The skills listed the elements of skills in the dataset are then divided into single elements that they may easily be compared with user skills. Lastly, any additional spaces and values that are unnecessary are eliminated to make sure that the information is highly organised and prepared to the recommendation process. This will enhance the precision and efficiency of the system.

D. Recommendation Technique: The proposed system applies a content based recommendation system to propose the course according to the interests and skills of the user. At first, the system displays only those courses which are related to the field of interest of the user. The matching process of the skills is done after filtering where the user skills are compared with the required skills in each course. The system identifies the relevance of a course by matching the skills of the user with the requirements of the course and scoring it based on similarity. Similarity score is done by the following formula:

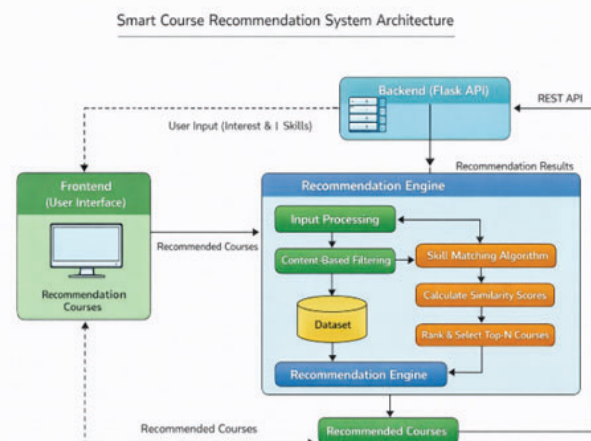
$$\text{MatchScore} = |\text{UserSkills} \cap \text{CourseSkills}|$$

Finally, the courses are ranked under a Top-N ranking scheme wherein the scheme takes the courses with most similar scores, and recommends one that would be the most practical to the user. This process ensures that the courses suggested are as close to user interests as well as skill set.

E. System Architecture Overview: The system gathers information about the user in terms of skills and interests domain.

Such inputs reflect the decision of the student and level of his or her knowledge at the moment. Once the user has entered this information, the system takes this information and compares it against the course information, which consists of a large and detailed amount of information about various courses, including course name, the level of difficulty, and type of interest among others. The primary approach utilized by the system is anchored on a content-based recommendation algorithm that compares the user preferences and course attributes. The system finds courses on the area of interest chosen by the user and then analyses the courses based on a skill matching algorithm. The algorithm computes the similarity between the skills of the user and the skills needed in each course of the dataset. A similarity score is produced in accordance with the number of shared skills.

Once similarity scores have been calculated the system then employs a ranking system to rank the courses in terms of relevancy to each other and then the courses ranked as most to least relevant are presented. The system then selects the most relevant courses to the user using a Top-N recommendation technique and provides the user with a list of the most relevant courses.



The system proposed is a web-based application on a client-server system. Users can enter their skills and interests into the front-end interface, and the backend server will process the information and give them suggestions. The backend is built with the Flask framework, and the course dataset is managed with data processing libraries. RESTful APIs are used to connect the front end and back end.

In order to improve the interpretability of the Smart Course Recommendation System, SHAP (SHapley Additive exPlanations) was deployed to identify the input of individual variables in predicting the results of the model. The outcome of SHAP results allows seeing the impact of the various course attributes on the recommendation and prediction results.

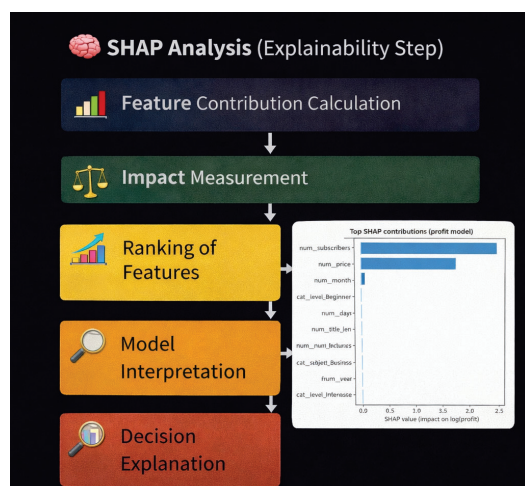


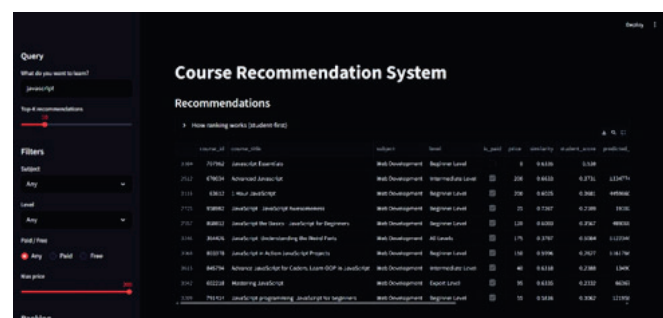
Fig. 1: It tells which features influenced the recommendation and by how much?

Overall, the goal of the system is to make the decision process easy for students by giving them intelligent and personalized course recommendations. The system helps students find the best courses for their learning path and career development by looking at what they give as input and course data. As the system architecture is dynamic in nature, we can easily add the new courses and skills in the future based on the technology that will be in trend.

IV. RESULTS

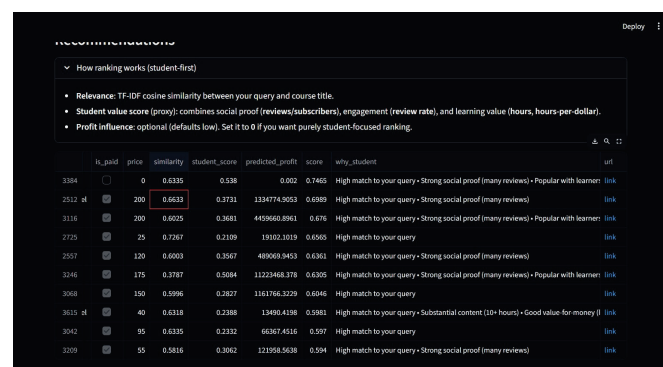
The proposed system web interface is a user-friendly interface. It consists of an input quest field and a recommend button. The user enters a course name in the quest field and clicks the recommend button also, the system analyzes the input title with all the titles in the dataset. Compare the input title to all the titles that are available, and then make a new dataset with the titles that match. Now among all the courses in the list, the courses with the highest recommendation score in descending order are suggested to the user. The proposed Smart Course Recommendation System was tested using a course dataset containing multiple domains such as artificial intelligence, web development, and data science. The system evaluates user input based on interests and skills and generates personalized course recommendations.

The experimental results showed that the system could successfully make personalized course suggestions. Let's say, when a user said they knew python, data analysis, and data science, the system suggest courses in data science, data analytics, and machine learning. When a user entered skills related to HTML, CSS, and JavaScript, the system also suggested courses in web development and Front-end technologies. This shows that the recommendation algorithm works well to find courses that are relevant by matching skills.



The results shows that the system filters relevant courses using content-based filtering successfully and matches user skills with course requirements accurately by using the skill-matching algorithm. The similarity score calculation helps the system suggest the best courses to the user by ranking them well. The results that are observed by experimenting indicate that the system:

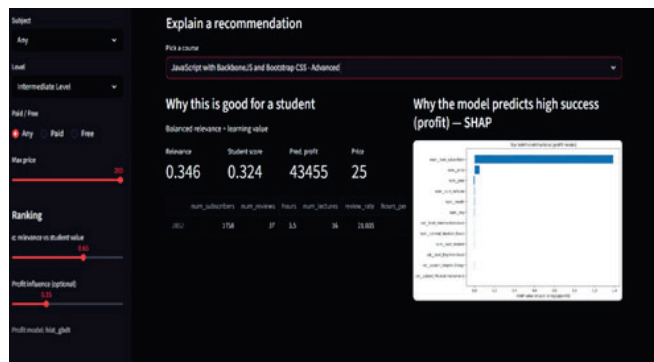
- It provides relevant and personalized course recommendations.
- It helps in reducing the difficulty faced by students in selecting appropriate courses based on interests and skills.
- Aligning courses with users' skills and interests makes it easier to make decisions.
- It efficiently processes user input and generates recommendations within no time.



This figure shows the results of the recommendations provided by the system through a student-centered ranking system. It shows several characteristic features of price of the course, similarity score, student score, predicted profit, and total ranking score. The similarity value (highlighted) shows the extent to which the course follows up the query of the user through the TF-IDF cosine similarity. The student score indicates the quality of the course in terms of the engagement metrics which include reviews, subscribers, and content value. The overall ranking score is calculated based on the sum of relevance, student value and optional profit influence, making sure that highly relevant and high-quality courses are given a priority. Also, the column of why student contains decipherable information about why this course or that one is advisable, which increases the level of transparency

and faith in the users.

The evaluation demonstrates that the proposed approach enhances the learning experience by guiding students toward courses that match their career goals and existing skill sets. The results confirm that the system is effective and can be further extended with larger datasets and advanced recommendation techniques.



The analysis of the results indicated that the number of subscribers, course ratings, and the number of reviews are heavily positive contributors to the success of a course as predicted. The more popular and well-rated courses will have more importance scores, which means they will have a greater role in the process of making a recommendation. However, the factors like price and course duration indicated the different degrees of impacts in varied situations where moderate pricing and course length were positively related to the model forecasts. The SHAP summary plots showed the general significance of features in all the predictions where popularity-based features take the center stage in the decisions made in the recommendations. Moreover, the explanations of SHAP values on an instance level presented instance-level information, indicating the reasons why a given course was suggested to a particular user. Finally, the results obtained after performing all this states that the Smart Course Recommendation System is capable of providing accurate, efficient, and personalized recommendations that are needed by the student. The recommender system helps students explore various learning opportunities and supports them in making correct decisions about their educational and career paths.

V. CONCLUSION

In this research paper, a Smart Course Recommendation System has been proposed to address the challenges students face in making the right decisions in the face of a choice of many courses which are available. Several students would face difficulties to find out courses that align with their interests and skills they already have to succeed in their academic performance and career growth. The given system is expected to make this process easier, as it will offer the user-specific course recommendations on the basis of inputs.

The results show that the suggested system makes it easier for students to choose from many course options. It helps people

make good decisions about school and work by suggesting classes that fit their interests and skills. The system also makes sure that data is preprocessed correctly and that the system architecture is well-structured so that processing is quick and recommendations are correct.

The system employs content-based filtering, which examines the interests of the users and compares it with the course data. Moreover, a skill matching algorithm is applied to assess the closeness of the skills that the user avails and the skills that are needed in a given course. The intersection of these skills is then calculated to determine a similarity score and the Top N ranking process is then employed to identify the most useful courses. This process would ensure that the prescribed courses are similar to the profile of the user by the system. This proposed system was implemented with the application of a web-based structure.

The frontend interface accepts user input and then the backend processes the input with an API that is based on Flask. The information is contained in the dataset that includes information on courses, such as the name of the course, the area of interest, the skills required, and the level of difficulty, which is used to make recommendations. To ensure making the recommendation process more precise and effective, we applied the appropriate data preprocessing tools, i.e., data cleaning, standardization, and skill extraction. The experiment outcomes demonstrate that the system is able to provide valuable and individualized suggestions in a relatively short period of time. It assists students in locating the appropriate courses and it also provides them with an avenue to organize their future careers in a methodical manner. This system can also be scaled to a very large extent due to the fact that additional new courses and skill categories may be introduced into the dataset and not in any way interfering with its general working. Moreover, the suggested system makes its contribution to the sphere of recommendations system in education through the combination of recommendation methods and easy-to-use web application. It also assists students to select the most suitable learning opportunities and have improved decisions concerning school work.

The system would be improved in future by providing collaborative filtering algorithms, machine learning algorithms, and live user feedback to give recommendations even more accurate. Besides, the system can become smarter and more skillful at making individual advice through fusion larger and more diverse data and designing adaptive learning models. Such improvements would expand the system's applicability in modern e-learning platforms and educational guidance systems.

REFERENCES

- [1] P. Resnick and H. R. Varian, "Recommender Systems," Communications of the ACM, vol. 40, no. 3, pp. 56–58, 1997.
- [2] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734–749,

2005.

[3] F. Ricci, L. Rokach, and B. Shapira, Recommender Systems Handbook, Springer, 2015.

[4] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender Systems Survey," Knowledge-Based Systems, vol. 46, pp. 109–132, 2013.

[5] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative Filtering Recommender Systems," Foundations and Trends in Human–Computer Interaction, vol. 4, no. 2, pp. 81–173, 2011.

[6] A. Manouselis, H. Drachsler, K. Verbert, and O. C. Santos, Recommender Systems for Learning, Springer, 2013.

[7] S. K. Mohanty, M. Jagadeesh, and H. Srivatsa, "Educational Data Mining: A Review of the State-of-the-Art," IEEE International Conference on Teaching, Assessment and Learning for Engineering, 2018.

[8] W. McKinney, "Data Structures for Statistical Computing in Python," Proceedings of the Python in Science Conference, 2010. (Related to the Pandas library used in the system.).

[9] M. Grinberg, Flask Web Development: Developing Web Applications with Python, O'Reilly Media, 2018.

[10] N. Yanes, A. M. Mostafa, M. Ezz, and S. N. Almuayqil, "A Machine Learning-Based Recommender System for Improving Students Learning Experiences," IEEE Access, vol. 8, pp. 201218–201235, 2020.

[11] H. Ahmadian Yazdi, S. J. Seyyed Mahdavi, and H. Ahmadian Yazdi, "Dynamic Educational Recommender System Based on Improved LSTM Neural Network," Scientific Reports, 2024.

[12] P.V Nagavardhan, B. Premamayudu , Published -2020 .Computer Science, Business. Recent Trends in Hybrid Recommendation Systems in e-commerce Domain.

[13] A collaborative filtering recommendation framework utilizing social networks Volume 14 , 15 December 2023.100495.

[14] Skill-based Career Path Modelling and Recommendation ,Aritra Ghosh, Beverly Woolf, Shlomo Zilberstein, Andrew Lan College of Information and Computer Sciences, University of Massachusetts Amherst