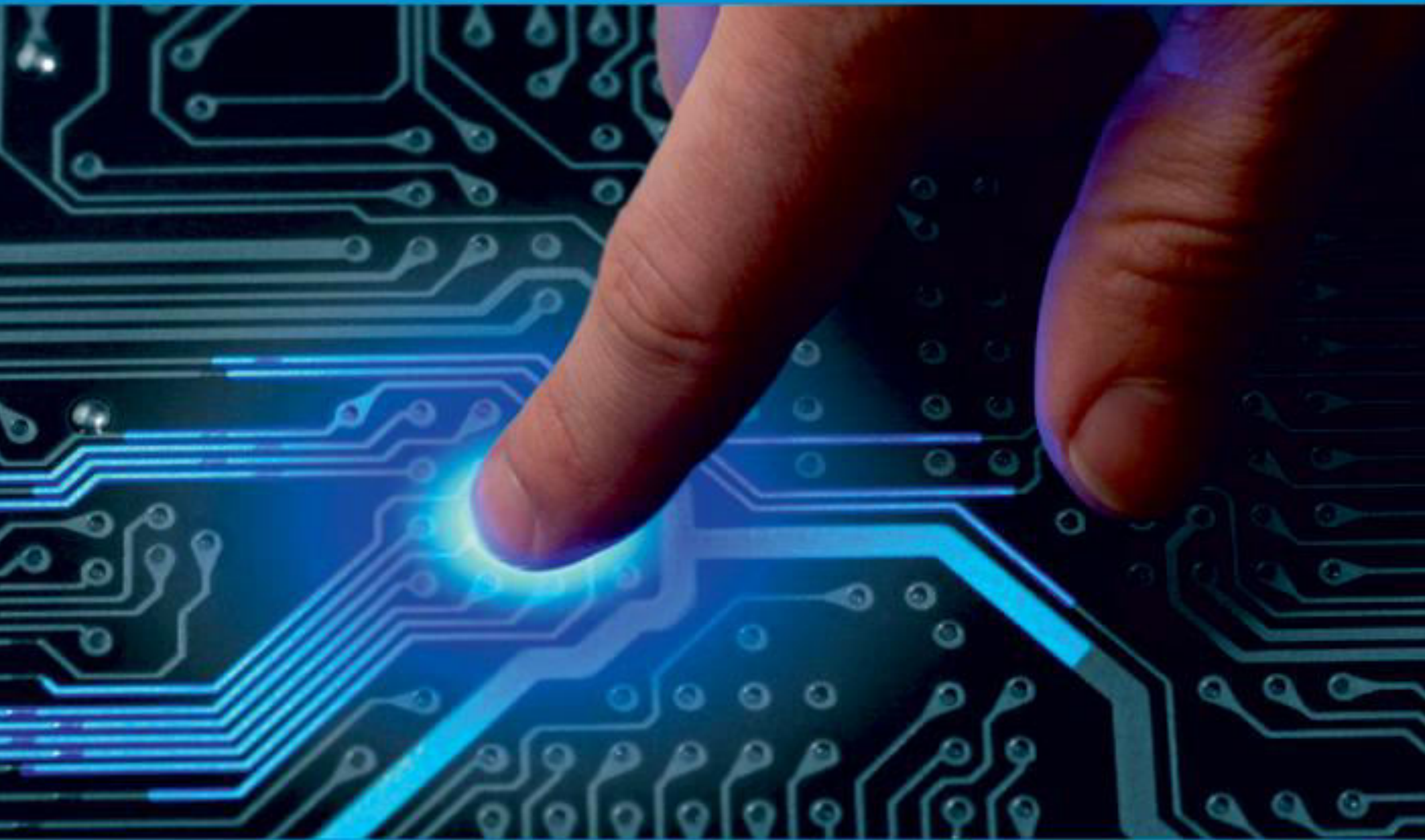
IJIRCCE

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



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Issue 5, May 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105192 |

EduNexus: A Personalized Course Recommendation System

[1]Neha Bhadale, [2]Bhakti Sable, [3]Sagar Deshmukh,[4]Vaishnavi Pawar,[5]Asst.Prof. Rutuja Kulkarni

[1,2,3,4,5] Department of Computer Engineering, Pune Institute of Computer Technology Pune, Maharashtra, India

ABSTRACT: The development of online learning platforms has given students access to a wide selection of courses on many subjects. For students, the sheer number of options can be overwhelming, and picking the right course can be challenging. In order to discover the course that best suits their interests, learners spend countless hours visiting each of the online course platforms. Thus, for finding courses that are a good fit for their interests and learning goals can be made easier for students by using personalized course recommendations. The proposed system is intended to provide tailored course recommendations based on the user's career interests, skill sets, and personal preferences. According to a learner's preferences, career interests, and skill sets, our system uses machine learning algorithms that can make useful cross-platform course recommendations in this study.

KEYWORDS: Recommendation system, Machine learning, Feature extraction, Data Mining

I. INTRODUCTION

Due to the global digital transformation, the education sector has made strides in recent years to adopt e-learning or online learning. Since the Covid-19 outbreak, adoption has accelerated even more, which has led to an explosive increase in online learning. People are exploring alternatives to traditional academic institutions and beginning to appreciate the flexibility of online education. These days, there are numerous online platforms that offer classes in a wide range of subjects and languages. They encourage the idea that anyone, anywhere, at any time, can learn anything with an internet connection.

Due to the abundance of options and learning platforms, however, curious students looking for online courses frequently experience option fatigue. This frequently results in a longer search period and greater search effort to locate the most appropriate course to begin learning. To address these inefficiencies, our team has designed an online course recommender system that seeks to be a one-stop platform. Our algorithm can make meaningful cross-platform course recommendations based on a learner's preferences, skill set, and career interests. Students would no longer have to spend countless hours visiting all of the online course platforms in search of the course that best suits their requirements.

Data collected from several course platforms is used to build the knowledge base for our system. We used technologies like Python and web-scraping techniques to extract course data from the platforms while creating the knowledge model. A SQL database is used to represent and store the knowledge base. Within our system, a recommendation reasoning system is being developed that can perform cross-platform course recommendations using a unique content-based filtering strategy. Using techniques like HTML, CSS, and JavaScript, a web framework is used to build the frontend user interface. We used the Python Flask app to manage the routes and requests across the system in order to link the system's frontend and backend, including the database and recommendation reasoning system.
.

II. LITERATURE REVIEW

Y. Ren, Z. He and T. Han; The research provides a better design for an online education course recommendation system, which includes data gathering, pre-processing, feature extraction, and the recommendation process. The proposed design attempts to improve the recommendation system's quality and the user experience. They propose a hybrid recommendation algorithm for course recommendation systems in online education. To overcome the limits of

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105192 |

each strategy and improve the accuracy of the recommendations, the proposed algorithm combines collaborative filtering and content-based filtering strategies.[1]

N. N. Y. Vo, N. H. Vu, T. A. Vu, Q. T. Vu and B. D. Mach, they proposed a system that would create personalized course recommendations for software engineering students by combining content-based filtering and collaborative filtering techniques. The authors gathered information on students' academic performance, course enrollments, and course content evaluations to create a knowledge graph that depicts the links between courses and skills. The CRS makes course suggestions based on this knowledge graph and a hybrid recommendation algorithm that matches students' talents and interests. The CRS can help students make more educated course enrollment selections and improve their learning results.[2]

Jiang, Pardo's, and Wei suggest a Goal-based course recommendation system (GCRS) that uses learners' goals and preferences to suggest courses. The work entailed gathering information from a significant online learning platform and developing a recommendation engine using this information. To find courses that were pertinent to the learner's choices and goals, the authors combined collaborative filtering with content-based filtering. They also included a goal model, which allowed students to identify their educational objectives and get tailored course recommendations based on these objectives. The authors discovered that their algorithm was particularly good at suggesting courses to students who had certain learning preferences and goals. In order to increase the usefulness of the recommendations made by the GCRS, the study also emphasized how crucial it is to provide students the freedom to identify their learning preferences and goals.[3]

Gulzar, Deepak, and A suggests a personalized course recommender system (PCRS) that makes course recommendations to students by combining collaborative filtering and content-based filtering. The authors contend that this strategy can increase the precision and applicability of the advice given to students. The authors utilized a content-based filtering algorithm to suggest courses that were appropriate for the learner's interests and skill levels and a collaborative filtering method to find courses that were popular among users who shared similarities. To increase the accuracy of the PCRS's suggestions, the authors additionally used a rating normalization technique. The authors discovered that their algorithm was especially good at suggesting courses to students with different backgrounds and interests. The study also emphasized the significance of including both collaborative and content-based filtering for creating tailored course recommendation systems.[4]

Zhao and Pan suggest a collaborative filtering algorithm-based methodology for recommending online courses. To find courses that were pertinent to the learner's interests and preferences, the authors employed an updated collaborative filtering algorithm that took into account a user's implicit input, such as click behavior and time spent on a particular course. To guarantee that the suggested courses were varied and not overly identical to one another, they additionally included a variety constraint. The authors discovered that their method was very good at suggesting courses to students with a variety of interests and preferences. The study also emphasized how crucial it is to incorporate implicit user feedback and diversity restrictions when creating course recommendation models.[5]

Jiang, Feng, Niu, and Dai suggests an intelligent recommendation system for online video courses. The authors contend that this strategy can enhance user interaction and engagement with online learning platforms. The work entailed gathering user information from a significant online learning platform and creating a recommendation engine using this information. To find courses that were pertinent to the learner's interests and preferences, the authors used collaborative filtering and content-based filtering. In order to assess the video content and make more accurate recommendations, they also included a deep learning algorithm. The authors discovered that their method was especially useful for suggesting courses to students with various backgrounds and interests. The study also emphasized the need of including deep learning models for creating knowledgeable recommendation systems for online video courses.[6]

.

III. METHODOLOGY

The proposed methodology consists of following main steps:

i. **Data collection:**
We used the data mining technique called web scraping to collect the course information because all of the course content is available online and accessible to the general audience. For conducting the web scraping on the twelve online course platforms, we used Python together with a number of auxiliary tools like BeautifulSoup and Selenium. Due to the differing complexity of web page design and structure across the twelve platforms, the web scraping procedure is carried out individually. To accommodate the various

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105192 |

html designs and structures of the online course platforms, the web scraping application written in Python is executed separately. Before being integrated and used in the following stages of knowledge modelling, the extracted course data from each platform is temporarily saved in the comma separated values (CSV) table format. The entire course data that we gathered fell under the field of computer science.

ii. **Data Pre-processing:**

With the raw course information extracted, we performed knowledge discovery and merged the datasets from the twelve online course platforms. In order to evaluate the importance and usefulness of the data collected by web scraping, we looked at the data types and contents of each data field during the knowledge discovery process. We noticed that the data gathered for each of the platforms has a variety of problems that need to be addressed. Missing data, inconsistent data values across platforms, and data values in various measuring standards are a few instances of problems. To maintain data consistency, several data cleaning methods are carried out, such as eliminating or replacing empty data fields with zero values and aligning the data type across the database

Below is a summary of three adjustments and transformations performed:

- Encoding course duration into 3 categories (short, medium, long) to ensure consistency across platforms.
- Encoding course difficulty into 3 categories (introductory, intermediate, advanced) to ensure consistency across platforms.
- Deriving a popularity index based on course ratings, by normalizing the number of ratings for each rating value and using the normalized ratings to calculate a weighted average rating. The popularity index is then calculated as the product of the normalized rating and the rating divided by the weighted average rating.
- Limiting course recommendations to only courses conducted in English

iii. **Feature Extraction:**

1. Feature Selection:

Feature selection, which is the process of choosing the most relevant and useful features (or qualities) from the preprocessed dataset for creating the recommendation system, would come after data preprocessing. The features are divided into two categories in this step: text-based features and numerical features. The choice of features is based on how courses are often chosen by a human learner, which enables the reasoning process to successfully carry out recommendation reasoning that is catered to or similar to a human learner's natural process of course selection. The feature selection process is crucial to reducing the dataset's complexity and enhancing the effectiveness and precision of the recommendation system.

Numerical Features	Text-Based Features
Course Duration	Course Title
Course Difficulty	Course Short Description
Course Free Option Availability	Course Long Description
	Course Categories

2. Text preprocessing of Text-based Features using NLP:

In this step, we have applied some NLP methods to extract and transform the data into appropriate feature representation that will be useful for the recommendation module. Following are a list of text pre-processing processed on each of the selected text-based features

Removal of Non-ASCII characters: As in the context of this project, we are only working with English language courses, all non-alphabetical and non-numerical characters are removed.

Lower Casing: Lower casing all alphabetical characters

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105192 |

Removal of White Spaces: Removal of extra and unnecessary white spaces before or after the text.
Removal of Punctuations: Removal of punctuations from the string which will not be useful for recommendation reasoning.
Tokenization: Convert the strings into list of tokens
Removal of Stop Words: Removal of stop words from the list of tokens which will not be useful for recommendation reasoning.
Lemmatization: Convert the tokens into lemma which is the basic forms of word.

3. Keyword extraction:

In this step, keywords are taken from the course description. Based on the Natural Language Processing (NLP) library spacy, the keyword extraction technique was built. Tokenizing the text, removing stop words and punctuation, and calculating the frequency of each word are all done using Spacy. The words are then ranked according to their frequency, and a list of keywords is returned. The ranking is determined by dividing each word's frequency by the highest frequency of all the terms in the text. The frequency of each word in the text is determined using the Counter function from the collections module in this straightforward frequency-based strategy for keyword extraction.

4. TF-IDF Vectorization (Converting a Bag of Words into a Sparse Feature Matrix):

After applying keyword extraction, we combined the four data fields that were produced into one list of tokens, which is essentially a collection of words. They are prepared for NLP-based feature extraction now that the combined tokens have been organized into a bag of words. On the collection of words, we used the Term Frequency-Inverse Document Frequency (TF-IDF) Vectorization. This converts the information into a sparse feature matrix that captures the text data's normalized word frequency information. A helpful feature representation that will be used in recommendation inference is the word frequency data. Every time a recommendation inference is performed, the sparse feature matrix is prepared for usage by further serializing it and storing it in the Python compact binary format known as Pickle.

5. One-Hot Encoding Features

We used a straightforward one-hot encoding to transform the numerical features from the course data and user inputs into binary matrices and vectors so they could be processed in the recommendation module.

iv. RECOMMENDATION SYSTEM:

1. Content-Based Filtering

Content-based filtering is a recommendation technique that uses the characteristics or content of items to make recommendations. It relies on the idea that if a user likes certain items in the past, they will likely be interested in similar items in the future. This algorithm analyzes the characteristics and features of courses, as well as user preferences, to generate recommendations. It calculates the similarity between courses based on attributes such as course topics, duration, difficulty level, and availability. Courses with similar attributes to the ones preferred by the user are recommended. The online course recommendation system utilizes content-based filtering with TF-IDF (Term Frequency-Inverse Document Frequency) and cosine similarity to generate personalized course recommendations. It is two-step Recommendation Processing:

Similarity Score and Ranking Optimization.

a. Similarity Function Module:

The computation of the similarity score using the cosine similarity function is the first step in the recommendation module. The cosine similarity of two vectors of length n is defined as follows:

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105192 |

In the TF-IDF sparse matrix, the similarity score between the user-input TF-IDF vector and each of the course feature vectors is calculated. At this point, a similarity score is calculated between the user-input one-hot encoded vector and each of the course feature vectors in the one-hot encoded matrix. However, only a portion of the one-hot encoded vector and matrix on course duration and difficulty is concatenated and used in this instance for the computation of the similarity score. The one-hot encoded vector and matrix from the course's free choice availability are not used in this case.

b. Ranking Optimization

The ranking optimization process, the second stage in the recommendation module, is a tailored ranking approach that incorporates some elements of rule-based ranking and dynamic ranking. The two sets of similarity scores (TF-IDF vectors and One-Hot encoded duration and difficulty) computed in the first step, the one-hot encoded free option availability vectors, and the popularity index of each course directly retrieved from the database will all be inputs used in the ranking optimization.

The 5+1 phases of filtering and sorting make up the ranking optimization. These personalized ranking and sorting techniques are made to be as efficient as possible while searching for and organizing the highest rated or most popular courses while also taking course similarity into consideration when making recommendations.

IV. SYSTEM IMPLEMENTATION

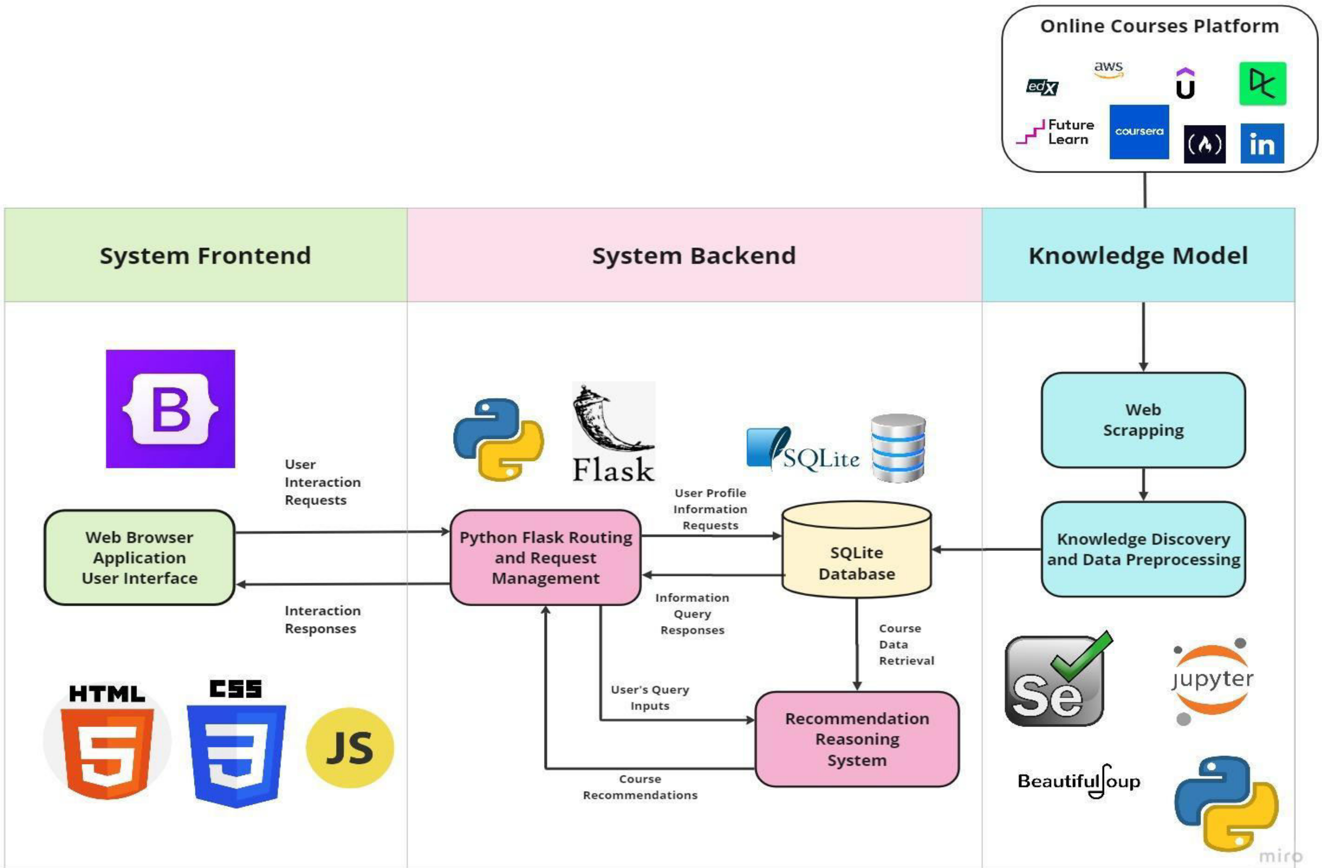


FIGURE : SYSTEM ARCHITECTURE

1. **Knowledge Model**

A cross-platform recommender system that is based on an extensive knowledge model has been created by us. Knowledge acquisition, knowledge discovery, and knowledge representation are the three main phases of the model. Using web scraping techniques, we gather information from multiple online course platforms during the knowledge acquisition step. This makes sure that we have a wide variety of courses from various sources.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105192 |

The knowledge discovery stage then involves preprocessing and analysis of the collected data. We correct inconsistencies, clean the data, and format it in a standard way. Finally, in the knowledge representation stage, we extract meaningful features from the course data. This includes information such as course titles, descriptions, duration, and difficulty level. These features are carefully selected based on their relevance to course recommendation.

2. **System Frontend**
The frontend user interface should be simple but useful. An HTML, CSS, and JavaScript-based web application serves as the framework for the system frontend. For interaction, the user interface is rendered in a web browser. Bootstrap templates, a popular modern web page design technique, are the CSS framework and style used for our frontend system. The backend request management module, which is based on the Python Flask application, is coupled with the frontend.

3. **System Backend**
Our system's backend consists of the Flask app module, a database serving as the knowledge base, and a recommendation system. The Flask app module manages request handling and routing between the frontend and backend. It is a flexible Python framework that allows for customization and integration of additional libraries. We utilize Flask-SQLAlchemy to interact with the SQLite engine and manage database queries. When a user requests course suggestions through the frontend, the recommendation system acts as the main engine for generating recommendations. The knowledge base, implemented as a SQLite database, serves as the foundation for the system. It comprises interconnected components representing our knowledge. Overall, our system combines the Flask framework, Flask-SQLAlchemy, and a recommendation engine to provide personalized course recommendations based on the user's inputs and the knowledge stored in the database.

V. RESULT AND ANALYSIS

We evaluated the performance of our content-based course recommendation system by analyzing the generated recommendations and comparing them with baseline methods. The analysis focused on the effectiveness of the system in providing personalized and relevant course recommendations based on preferences and career interests.To conduct our evaluation, we utilized a dataset comprising 12 cross-platform courses from various disciplines. We individually scraped the relevant data for each course, including attributes such as course title, description, rating, platform, no. of enrollment etc.

Our content-based course recommendation system demonstrated its ability to generate tailored recommendations by leveraging the course content attributes along with course preferences and career interests. The system matched the interests and preferences of users with courses that aligned with their given preferences, and desired career paths.The significance of our research lies in the potential to enhance the course discovery process for students by considering preferences and career Interest. Our system helps students explore a wider range of relevant courses that match their preferences desired career paths. This can lead to a more fulfilling and engaging educational experience, while also aligning with their long-term career objectives.

It is important to acknowledge the limitations of our content-based method. The accuracy and quality of the recommendations are dependent on the completeness and correctness of the course metadata, content attributes, and the information related to course preferences and career interests. Inaccurate or incomplete information may affect the relevance of the recommendations. Additionally, the content-based approach may face challenges in handling cold-start situations for new users or courses with limited data.

VI .CONCLUSION

In this paper, we proposed a personalised course recommendation system that uses machine learning techniques to provide course recommendations based on the learner's skills and interests. The suggested system collects data on the student's skills and interests, develops learner and course profiles, and uses a hybrid recommendation model to generate tailored course recommendations. The suggested method has the potential to increase learners' learning outcomes and engagement by giving tailored course recommendations that fit their abilities and interests. The system leverages content-based filtering, TF-IDF vectorization, and cosine similarity to provide personalized course recommendations to

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |
|| Volume 11, Issue 5, May 2023 ||
| DOI: 10.15680/IJIRCCE.2023.1105192 |

users based on their interests, skills, and learning goals. The system's frontend and backend modules have been designed and integrated, allowing for seamless user interaction and efficient recommendation inference. The evaluation of the system has demonstrated its effectiveness in assisting users in discovering relevant and suitable courses, ultimately enhancing their learning experience.

VII. FUTURE WORK

Although the course recommendation system has achieved its primary objectives, there are several avenues for future work and improvements:

- Integration of Collaborative Filtering: Collaborative filtering, which takes into account user behavior and preferences, can be incorporated into the recommendation system. By combining content-based filtering with collaborative filtering, the system can provide more accurate and diverse recommendations.
- Integration of Additional Data Sources: Currently, the system relies on data from specific online learning platforms and courses of Computer science domain. Integrating data from a wider range of platforms and diverse educational resources can enrich the recommendation process and provide users with a broader selection of courses.
- User Feedback and Iterative Improvement: Gathering user feedback, analyzing user behavior, and continuously refining the recommendation algorithms based on user input and evolving user preferences.

REFERENCES

[1] Y. Ren, Z. He and T. Han, ”Research on Optimal Design of Online Education Course Recommendation System Based on Hybrid Recommendation Algorithm,” in 2021

[2] N. N. Y. Vo, N. H. Vu, T. A. Vu, Q. T. Vu and B. D. Mach, ”CRS - A hybrid Course Recommendation System for Software Engineering Education,” 2022 IEEE/ACM 44th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET), 2022.

[3] Weijie Jiang, Zachary A. Pardos, and Qiang Wei. 2019. Goal-based Course Recommendation. In Proceedings of the 9th International Conference on Learning Analytics Knowledge (LAK19). Association for Computing Machinery, New York, NY, USA, 36–45.

[4] Gulzar, Z., A, A. Deepak, G. (2018). PCRS: Personalized Course Recommender System Based on Hybrid Approach. Procedia Computer Science, 125:518–524.

[5] L. Zhao and Z. Pan, ”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), 2021.

[6] P. Jiang, Y. Feng, C. Niu and Y. Dai, ”Study of intelligent recommendation for online video courses,” 2021 IEEE 5th Information Technology,Networking,Electronic and Automation Control Conference (IT- NEC), 2021, pp. 1290-1294.