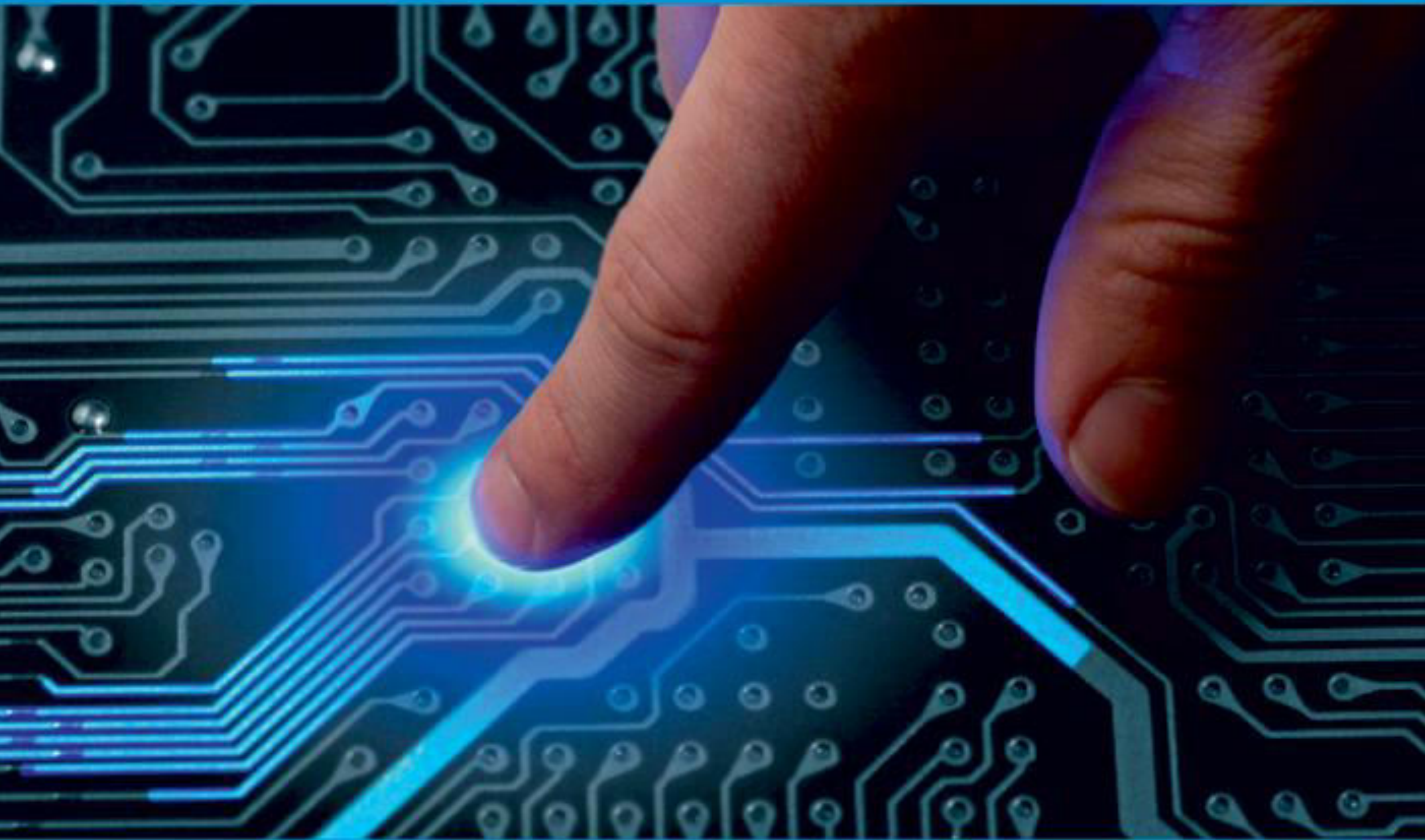
IJIRCCE

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.2023.1105351 |

Course-Placement Recommendation System Using Machine Learning

S Sabana Singha, Shalu Sharma, Shanu Kumar, Dakshayni Awate, Prof. Rohini Bhosale

Department of IT, MIT School of Computing, Pune, India

Department of IT, MIT School of Computing, Pune, India

Department of IT, MIT School of Computing, Pune, India

Department of IT, MIT School of Computing, Pune, India

Assistant Professor, Department of IT, MIT School of Computing, Pune, India

ABSTRACT: Since the epidemic of 2020, Online learning platforms have proliferated. Individuals worldwide are ready to learn and experience new things during their free time. Furthermore, in this new world of technology, content is available online for free but in large quantities. Because this freely available content is crowded, many people have difficulty accomplishing their goals with adequate free coaching. Speaking of E-Learning, it is primarily aimed at two types of students:

a) traditional people who only seek knowledge before entering the professional world, and
b) workers who have prior knowledge but need to improve what they already know to meet job requirements; this learning is referred to as lifelong learning.

Because of technological advancements, there are several learning resources available. Various sources for learning, e.g., for notes, there are numerous resources, and for learning from video sources, there are countless numbers of video content, which drives users to churn from one source to another and therefore leads to distraction from purpose without sufficient supervision. To address this, we developed a course recommendation system to assist learners in selecting the appropriate course.

Keywords: E-learning, Online course recommendation system, Course recommendation, learning analytics, Learning outcomes, Competency matrix, Sparse Linear Method, Smart Education, Collaborative filtering, Similarity, Non-formal education, Student skill.

I. INTRODUCTION

Much study has been conducted to improve recommendation systems for online learning; with this project, we intend to research and test various Machine learning algorithms and their combinations to get good outcomes. But first, it's critical to comprehend why recommendation systems were created. In today's world, where the internet is filled with a massive amount of information, it is extremely difficult to classify and decide which information is more relevant to the context or more accurate, so recommendations were created to filter out the extra clutter and produce the desired or similar results. In the case of E-learning, the internet is brimming with resources from which to learn, but it can be difficult to select the correct content or course that will assist them in attaining their objectives. Even websites that offer to assist in such instances frequently end up overwhelming learners with course suggestions, causing cognitive overload and making it difficult to determine what to learn and where to learn from. We approached this problem by researching various recommendation algorithms and their combinations, including the Random Prediction Method, Popularity Model, Demographic-based Filtering Systems, Knowledge-based Recommendation Systems, Content-based Filtering Systems, Collaborative Filtering Systems, and Hybrid Recommendation Systems; we then implemented these systems.

II. LITERATURE SURVEY

This study was designed with the goal of reducing the learner's effort and time spent searching for the proper course. Viet Anh Nguyen [1] performed a survey with the goal of developing a system that will recommend appropriate classes for each student in the forthcoming semesters based on their present academic performance. They employed data mining and learning analytics techniques to estimate students' learning outcomes for the upcoming semester and developed a model to pick the best courses for each student. They proposed that each course be treated as an item in a

| 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.2023.1105351 |

competency matrix, and that students' marks be treated as users rating the relevant items. We presume that the grades of each student are comparable, which explains their likeness. The User-Based Collaborative Filtering technique predicts a student's course grade based on similarities between students. Their course grades reveal how similar these students are to one another. With smaller score differences, the degree of likeness grows.
Sunita B Aher [2] compares various data mining algorithm combinations, such as clustering and association rule algorithms, association rule mining of classified and clustered data, combining clustering and classification algorithms into association rule algorithms, and using only association rule algorithms. They look at the AD Tree classification, Simple K-means clustering, and the Apriori association rule algorithms. They compare clustering and association rule algorithms, association rule mining of categorized and clustered data, integrating clustering and classification methods into association rule algorithms, and only association rule algorithms. According to their simulation, the optimum algorithm combination includes clustering, classification, and association rule mining.

Jing Li[3] investigates ways to bring personalized recommendation technology, which is widely used in industry, to online learning. The platform for personalized learning is then constructed and employed, which is based on a collaborative filtering algorithm. The personalized recommender system achieves its purpose by combining data from data processing services with the model from the model library, doing algorithm calculations in line with the algorithm formula, and then offering the things customers require. To locate similar users or items, i.e., nearby users, one must first compute similarities between computed people or objects. It then forecasts scores by averaging the scores of nearby users.

Huynh-Ly Thanh-Nhan [4], like Viet Anh Nguyen [1,] suggested a system with three key feature groups: grade prediction, data transfer, and course recommendation. The training/predicting programme was implemented as a desktop application, with the missing data features/values pre-processed. Upon training, the system moves the grading matrix table from the app-server to the webserver. Following prediction, all grades are saved in the grading matrix and sent to a web application for course suggestion.

Jinjiao Lina's paper [5] discusses They suggested a sparse linear-based technique for top-N course recommendation by including expert knowledge and sparseness regularization into the calculation. The method they proposed is primarily concerned with the correctness of course suggestions in comparison to actual data acquired from experts.

Another web-based system developed by Ko-Kang Chu [6] Actual course selection records for two classes over two academic years are collected using the course selection method. After the order of the students' choices was determined through their recommendation procedure, the most acceptable courses for suggesting learners could be chosen.

This collaborative recommender system was described in article [7] by combining collaborative filtering with the Creating a Curriculum (DACUM) Approach. The course is recommended to learners based on their knowledge level, learning skills, and learner profile. For this method, data from nine classes were collected. This recommendation system outperforms a standard collaborative filtering recommender by 0.6 mean value satisfaction.

The study [8] examined all types of recommended systems and highlighted their difficulties. They suggested a system that discovers similar students automatically and then applies an association rule mining technique to their courses to generate course association rules.

To obtain recommendations, discovered course association rules are employed.
We researched [9],[10], which were based on surveys and research on the student learning experience, because our main emphasis was to develop an easier learning experience for learners and most articles were focused on university students and how to suggest better courses. The SWOC analysis was performed in a study [9] by Shivangi Dhawan to understand numerous strengths, weaknesses, possibilities, and problems related with the online learning modality during this critical situation. The content analysis research tool was utilized to analyze the data gathered from various sources for this study, and the descriptive research approach was applied. Another study [10] linked online learning experience with pandemics. Data was collected and evaluated to identify bottlenecks in online learning, and ideas for overcoming specific obstacles were made, which aided in providing a solution not only during a pandemic but also in the long run. The ratings from available e-learning platforms were collected, and a net promoter score (NSP) was developed, which aided in the implementation of solutions with Psychological and biological elements considered. Furthermore, online learning is the best option available to combat the current scenario, but the development of such a platform must be improved in order to consider ways of learning entirely fruitful.

| 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.2023.1105351 |

III. HARDWARE AND SOFTWARE REQUIREMENTS

1. Hardware used-
    - Processor.
    - 40 GB Hard Disk
    - Min 256 MB RAM Requirements.
2. Software and libraries used-
    - Selenium
    - Pandas.
    - Sci-kit.
    - Nltk library
    - Plk library
    - H5py library.
    - Web Browser (Firefox, Chrome, etc.)
    - Any OS

IV. SYSTEM ANALYSIS

4.1. Existing System

For this research, we investigated current course recommendation systems, the bulk of which were focused on performance-based outcomes and made suggestions from either limited courses or courses offered by the institution. The most popular existing method is to collect data from students and then use it to anticipate better courses for them, although data obtained from users can be influenced by a variety of circumstances. This utterly ignores the reality that other elements, such as popularity, level, and so on, play a major role in course selection decision-making.

4.2. Proposed System

The proposed system is a course-placement recommendation system that aims to provide a more personalized and comprehensive solution to students. The system will combine both course and job placement recommendations on the same platform, making it easier for students to find the right course and job placement. The system will use machine learning algorithms to analyze the user's profile, interests, and learning history to provide personalized recommendations.

The proposed system will have the following features:

4.2.1. Personalized Course Recommendations:

The system will analyze the user's profile, interests, and learning history to recommend courses that are relevant to their interests and learning goals. The system will use machine learning algorithms to analyze the user's learning history to recommend courses that are relevant to their previous courses and learning progress.

4.2.2. Personalized Job Placement Recommendations:

The system will analyze the user's profile, interests, and job history to recommend job placements that are relevant to their interests and career goals. The system will use machine learning algorithms to analyze the user's job history to recommend job placements that are relevant to their previous job experiences and career progress.

4.2.3. Course and Job Placement Matching:

The system will match the user's recommended courses with relevant job placements to provide a comprehensive solution to the user's career goals. The system will use machine learning algorithms to analyze the user's profile, interests, and job history to recommend courses that are relevant to their desired job placement.

In today's world, where technology is advancing at a rapid pace, the education sector is also evolving. With the growing number of courses and specializations, it becomes difficult for students to choose the right course that will help them in achieving their career goals. This is where recommendation systems come into play. Recommendation systems are widely used in various e-commerce platforms, social media, and now in education as well. In this paper, we will discuss the process of creating a course and placement recommendation system using Python.

Creating Dataset:

The first step in creating a recommendation system is to gather data. In this case, we will be using web scraping 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.2023.1105351 |

gather data from various sources such as college websites, LinkedIn, Glassdoor, etc. We will be using the Selenium library to scrape the data and save it in CSV format using the Pandas library. Once the data is gathered, we will drop any duplicate data rows and clean up any commas, white spaces, etc.

Creating Model:

The next step is to create a model that will recommend courses and placements to the user. We will be using Text Vectorization to convert the text data into numerical data. We will be using the sci-kit learn library for this purpose. Once the text data is converted into numerical data, we will be applying the Porter stemming algorithm using the nltk library to reduce the words to their base form. This will help in reducing the data size and improving the efficiency of the model.

The next step is to find the similarity between the vectors using the cosine similarity measure. This will help in determining how similar the courses and placements are to each other. We will be using the sci-kit learn library for this purpose as well. Once the similarity measure is calculated, we will build a recommendation function that will recommend courses and placements based on the user input.

The next step is to export the model using the Python Pickle file (.pkl) for course recommendation and Hierarchical Data Format (.hdf5) for placement recommendation. We will be using the Pickle library and h5py library respectively for this purpose.

Creating UI:

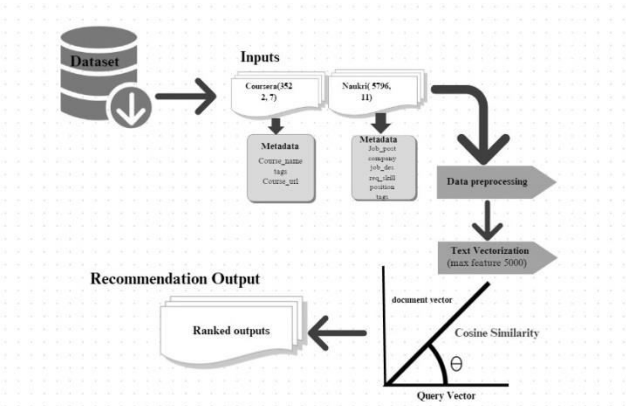
The next step is to create a graphical user interface using the PyQt5 library. The GUI will take input from the user and display the recommended courses and placements. We will be using the pre-trained models to display the recommendations.

Testing and Deployment:

The final step is to test the model with sample data to check its accuracy and efficiency.

The course and placement recommendation system can be evaluated using various metrics, such as precision, recall, and accuracy. These metrics assess the system's ability to provide relevant and accurate recommendations to students. Additionally, user feedback and satisfaction surveys can be collected to gauge the system's usability and effectiveness.

4.3. System Architecture Diagram



4.4. Methods

Case studies can be conducted to showcase the system's performance in real-world scenarios. Students with different backgrounds and career goals can use the system to find suitable courses and job placements, and their feedback and experiences can be analyzed to measure the system's success.

Recommendation systems have become an integral part of our daily lives, from suggesting products on e-commerce websites to recommending movies on streaming platforms. These systems are designed to provide personalized recommendations to users by analyzing their past behaviors and preferences, and suggesting items that they are likely to enjoy. One of the most popular approaches to building recommendation systems is using similarity measures, which help identify items that are like the ones a user has already interacted with. In this paper, we will discuss the cosine similarity measure, which is commonly used in recommendation systems.

Natural Language Processing (NLP) is a field of study that focuses on the interaction between computers and humans using natural language. It involves the processing and analysis of human language to derive meaning and

| 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.2023.1105351 |

insights. One of the key techniques used in NLP is cosine similarity. Cosine similarity is a measure of similarity between two vectors in an inner product space. In this paper, we will explore how cosine similarity is used in NLP, its advantages and limitations, and some applications of this technique.

Similarity measures are used to determine the degree of similarity between two items or users in a dataset. The most common similarity measures used in recommendation systems are:

1. **Euclidean Distance** - measures the straight-line distance between two points in a multi-dimensional space.

2. **Manhattan Distance** - measures the distance between two points by summing the absolute differences between their coordinates.
3. **Jaccard Similarity** - measures the similarity between two sets by comparing the number of common elements they share.
4. **Minkowski Distance** - measures the distance between two points in a multi-dimensional space using a generalized distance formula.
5. **Cosine Similarity** - measures the cosine of the angle between two vectors in an n-dimensional space.

Cosine similarity is a commonly used similarity measure in recommendation systems. It measures the similarity between two vectors in an n-dimensional space by calculating the cosine of the angle between them. The resulting value ranges from -1 to 1, where 1 indicates that the vectors are identical, 0 indicates that they are orthogonal, and -1 indicates that they are diametrically opposed.
The formula for cosine similarity is as follows:

$$\text{cosine\_similarity}(u, v) = (u \cdot v) / (\|u\| * \|v\|)$$

where u and v are the two vectors being compared, and $\|u\|$ and $\|v\|$ are their respective magnitudes.
In the context of recommendation systems, u and v can represent user-item interactions, where the values in the vectors represent the ratings or preferences of the user for the items. The cosine similarity measure can then be used to identify items that are like the ones a user has already interacted with and recommend them accordingly.

Cosine similarity has many applications in NLP. One common use case of it in the project is document classification. In this task, documents are classified into different categories based on their content. Cosine similarity can be used to compare the similarity between a given document and a set of pre-defined categories. The category with the highest cosine similarity score is then assigned to the document.

Another application of cosine similarity is information retrieval for the project. In this task, a user enters a query, and the system returns a set of documents that are relevant to the query. Cosine similarity can be used to compare the similarity between the query and the documents in the database. The documents with the highest cosine similarity scores are then returned to the user.
This is how the course recommendation system concept works, and users can use any way they choose to implement and expand the process as needed. For example, in the study article, we used streamlit to implement the workings of countVectorization and combine it with a web App.

V. RESULT & DISCUSSION

A course-placement recommendation system is a sort of software application that assists users in finding the best courses based on their interests, talents, and learning objectives. A course-placement recommendation system saves students time and effort by recommending appropriate courses based on their interests, experiences, and career aspirations. It can also help to improve learning quality by making customised recommendations based on the learners' needs and preferences. The constructed course-placement suggestion system is incredibly user-friendly and simple to use. When the website loads, the visitor will see a screen where they may enter any course name to get further information and recommendations.
When the course name is typed, a list of suggestions shows with various details such as the course link, score, and price.
After selecting the chosen course, students must click on the proposed course, which loads a new page where they can begin their trip.

| 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.2023.1105351 |

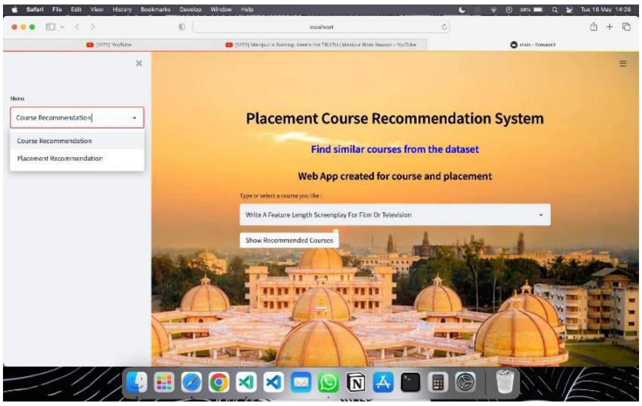


Image 1. UI Screenshot_1

Image 2. UI Screenshot_2

VI. CONCLUSION & FUTURE SCOPE

One of the most significant considerations that learners must make before beginning to learn a new skill or upskill is the selection of relevant courses. Our project aims to create solutions that offer appropriate and suitable courses for each student based on popular indicators such as ratings and subscription count.

The selection of relevant courses is one of the most important considerations that learners must make before beginning to learn a new skill or upskill. Based on popular measures such as ratings and subscription count, our project strives to develop solutions that deliver relevant and suitable courses for each student.

Even though the studies were carried out with a limited amount of data, the course selection based on the most popular courses shows some potential. We learned about how different aspects influence a student's decision-making when selecting courses in this project, as well as how data may be streamlined for better results.

The recommendations are based on a dataset of udemy courses, which can be mitigated by upgrading the dataset with fresh popular courses.

The user interface may not be appealing to all people, but we can always improve based on user feedback.

Future work will focus on expanding and enhancing the model, as well as other issues like starting to recommend courses that are more suited to individual requirements. Collecting more data about students (interests, talents, and needs) may also be taken into consideration. Additionally, we now lack the ability to feed datasets to models; however, that is something we hope to change in the future.

Our dataset may be expanded with new, well-liked courses and filters to sort and provide more individualised results for a particular user.

VII. ACKNOWLEDGEMENT

I express my profound thanks to my Guide Prof. Rohini Bhosale for her expert guidance, encouragement, and inspiration during this project work. I would like to thank Prof. Kanchan Katake, Project Coordinator, and Prof. Rohini Bhosale, Project Guide Department of Information Technology for extending all support during the execution of the project work. I sincerely thank Prof. Dr. Rekha Sugandhi, Head, Department of Information Technology, MIT School of Engineering, MIT-ADT University, Pune, for providing necessary facilities in completing the project. I am grateful to Prof. Dr. Rajneeshkaur Sachdeo, Director, MIT School of Computing, MIT-ADT University, Pune, for providing the

| 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.2023.1105351 |

facilities to carry out my project work. I am also thanking all the faculty members in the Department for their support and advice.

REFERENCES

[1] V. A. Nguyen, et al., "A course recommendation model for students based on learning outcome," Education and Information Technologies, Springer, vol. 26, no. 5, pp. 2021.
[2] S. B. Aher, et al., "Best Combination of Machine Learning Algorithms for Course Recommendation System in E-learning," International Journal of Computer Applications, vol. 41, no. 6, pp. 1-10, 2012.
[3] J. Li, et al., "Course Recommendations in Online Education Based on Collaborative Filtering Recommendation Algorithm," Complexity, vol. 2020, IF2.833.
[4] "COSINE DISTANCE, COSINE SIMILARITY, ANGULAR COSINE DISTANCE, ANGULAR COSINE SIMILARITY," [Online]. Available: www.itl.nist.gov. [Accessed: July 11, 2020].
[5] J. Lina, et al., "Intelligent Recommendation System for Course Selection in Smart Education," in 2017 International Conference on Identification, Information and Knowledge in the Internet of Things, 2017.
[6] G. Sidorov, A. Gelbukh, H. Gómez-Adorno, and D. Pinto, "Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model," Computación y Sistemas, vol. 18, no. 3, pp. 491-504, Oct. 2014.
[7] "Cosine similarity," Wikipedia, last edited on Feb. 26, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Cosine_similarity. [Accessed: Feb. 26, 2023].

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.379

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING