BioGecko

Vol 11 Issue 03 2022
ISSN NO: 2230-5807

A Customized Recommendation System using Streamlit

Hemalatha Eedi[1]
Assistant Professor, Department of CSE, JNTUH University College of Engineering Hyderabad
hemamorarjee@jntuh.ac.in
Morarjee Kolla[2]*
Associate Professor, Department of CSE, Chaitanya Bharathi Institute of Technology
morarjeek_cse@cbit.ac.in

Abstract: Nowadays several learning courses are available online and selecting a suitable course among them as per user requirements is a daunting task. For users to achieve their desired goals, picking the right course is crucial. It can be challenging for them to pick a course that fits their preferences, goals, and present knowledge. This paper aims to construct a prototype system to choose online courses based on the user's log. Our system uses collaborative filtering techniques to extract the information and suggest courses. This technique is used to measure the similarities and suggest recommendations of user interest. Our system is used to find the utility information from the utility logs of other users. Item-wise comparisons using cosine similarity of two items computed. Our proposed system finds the novel interest of the users and also fetches similar users interested courses. In this paper, a relevance factor is calculated based on which courses are recommended to the user. Our system uses Streamlit for deployment and produces optimal recommendations compared to competitive models.

Keywords: Collaborative Filtering, Course, Learning, Recommendations, Streamlit, Utility.

1. Introduction

Smarter decisions are made as a result of having more information, yet occasionally too much information affects the quality of our decisions [1]. Users will never be able to quickly identify the information they are interested in from the sea of data. By reducing some forms of information overload, recommendation systems can improve the adaptability and effectiveness of learning environments in the educational setting. It could be challenging for a user to choose a course among the many options accessible. A software called Recommendation Systems (RS) has come up with a solution to handle this issue [2].

This paper's scope encompasses a course recommendation system that will make use of cosine similarity to provide personalized recommendations. In the recommendation system, the problem is trying to forecast the opinion the users will have on the dissimilar substance and be able to recommend the finest items to each user [3]. A more effective recommendation system that shortens the system's runtime and more accurately detects item relations while improving the attributes of existing systems.

Recommending the movie or books based on their category works mainly uses traditional recommender systems. Most of the researchers suggest recommendations based on the content or the group of similar users' interests [4]. These traditional approaches are purely depending on the association among the items or users. Hence these traditional approaches limit the finding of original user interest that might be changing from time to time. Sometimes based on historical, local, trendy, traditional, influential, and attractive situations may lead to an increase the interest in some novel pattern that may not be interested in the user earlier. Surprisingly time users may show interest in a

novel pattern that was not matched with earlier interested patterns [5]. Extracting these nontraditional recommendations from the system is a crucial task nowadays.

Recently conversational recommendation systems grab the attention of researchers to predict relevant recommendations with greater quality [6]. These recommendations need quality datasets to extract crucial features. Traditional recommendation approaches are still in the infant stage. To overcome these difficulties advanced deep learning-based approaches with high-level features produce prominent results with Convolutional Neural Networks (CNN) [7]. In the real-time scenario, most of the applications are web-based for recommender systems. To develop web apps, Streamlit is user-friendly for model deployment and finetuning [8]. This proposed recommender system customizes using Streamlit for high-quality recommendations [9]. The main novelty in this work is customized recommendations for individual users with high quality. Many real-time applications for recommending semantic items extract knowledge from collaborative users [10].

2. Related Work

In the area of Recommender Systems, many studies have been conducted. New algorithmic methods are being proposed by researchers, and Recommender Systems are being used in practical application domains. The analysis of students' course selection behavior has been the subject of numerous studies in the area of online learning.

A learning management system was suggested by Imran [11] et al. as one of the technologies to be used in a personalized recommendation system. The conventional algorithms merely take into account user ratings and ignore changes in user interest and the veracity of rating data, which have an impact on the effectiveness of the system's recommendation. As a result, the paper offers a better approach to address this issue. The central concept is predicated on the notion that similar consumers will have similar preferences. by using a user similarity algorithm based on user reviews to identify nearby individuals who share the same interests as the active user. By adding the neighbor's ratings for the same item, it is then possible to forecast the active user's preference. Lastly, the top-N courses that the active user will probably enjoy are offered. However, the collaborative filtering algorithm's similarity measurement only considers the similarity score and ignores user interest.

The current education system in India needs automation support to facilitate students for various academic activities. This support system provides facilities for online mentoring, course support, teaching, and learning [12]. Due to technological improvement day to day significantly increases the new courses. Now a large number of courses with different contents are available globally. Several Universities and colleges are offering many courses to the students. Each course is designed with different objectives and outcomes. Selecting a suitable course with willing content is a crucial task for students nowadays. Manually running this system with these many courses is a difficult task. Automation of this process is the only solution, which can counsel, recommend, and guide the students in several aspects of the teaching-learning process. Web recommender systems are useful nowadays with greater user interfacing services and support [13].

Current course recommendation systems need personalization support for various academic activities. Selection of the course depends on the branch, continuity of courses, the relationship among the courses, the outcome of the course, the demand of the course, knowledge acquired from the course, and opportunities available in the market for that course [14]. Online recommender systems should consider the career planning of the student and also other preferences of the student like research, entrepreneurship, and higher education. Recent research in the course recommendation systems are focusing on developing quality system with optimization [15]. These high-quality recommender

systems with web support are developed using advanced software frameworks. Our proposed system addressed the current challenges of web-based recommender systems.

3 Course Recommender System using Streamlit

Our proposed course recommender system follows the flow as mentioned in the flow diagram. Basic algorithm steps followed to recommend the course are given step by step.

3.1 Course Recommender Algorithm

Our Proposed system follows the steps mentioned below

Step 1: Initially, import all the necessary libraries and also import the DataSets.
Step 2: Integrate all the datasets from different platforms, and clean the datasets by using pandas.
Step 3: Integrate all the attributes into a single attribute named “tags”.
Step 4: Convert the words into root words using porterstemmer().
Step 5: Tokenize the words using CountVectorizer.
Step 6: Using streamlit create a web application with a Home, Recommend, and About Page.
Step 7: From the recommend page take input from the user and pass it on to get_recommendation().
Step 8: If the course is found, distances between input and other courses are calculated based. on key attributes in the dataset using cosine Similarity distance.
Step 9: Otherwise, courses are generated based on the terms present in the input.
Step 10: Based on the cosine similarity distances similar courses are recommended.

3.2 Flow Diagram of Proposed Course Recommender System

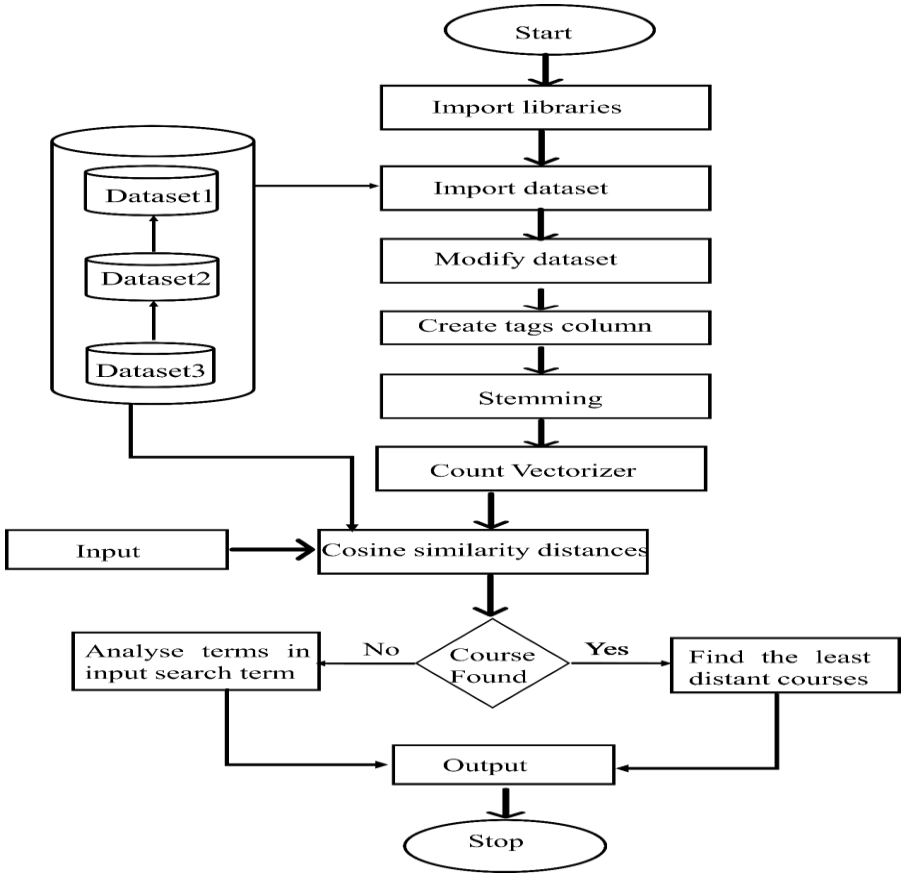

Figure 1: Flow diagram of Proposed Course Recommender System

The flow diagram shown in Figure 1. depicts the step-by-step process from loading the dataset to generating customized recommendations. Preprocessing of input with Stemming and Counter vectorizer and measuring the similarities of users to find the suitable course.

4. Implementation Details

The language on which we built our software was Python. It makes use of several Python libraries.

4.1 Python Libraries: The following Libraries, Frameworks, and Tools are used to develop the Proposed system.

Pandas: Open-Source library to work with labeled data.
Sklearn: Open source library for Machine Learning
NLTK: Package for Natural languages
Seaborn: Library for visualization
Streamlit: Open Source Framework for Machine Learning and Data Science web apps.
PIL: Library for Image Processing

Some of the crucial functions used in implementation along with their functionalities are listed below.

st.write(): The Streamlit **st.write**() function is a powerful tool and can be used in many different ways, from writing ordinary text to displaying figures.
st.markdown(): Markdown has several uses within a streamlit application. As the only tool for custom HTML within a Streamlit app, we can use it to flexibly insert rich content into your application.
St.sidebar: It is a simple Streamlit component that allows users to organize elements into a sidebar.
st.header: It displays text in header formatting.
st.subheader: It displays text in subheader formatting.
st.text: It is used to write fixed-width and preformatted text.
st.info: It displays an informational message.
st.warning: It displays a warning message.
St.error: It displays an error message.
st.image: Display an image or list of images.
st.button: It displays a button widget
st.data frame: It displays a data frame as an interactive table.
stc.html: HTML compress for stc, support template syntax.

4.2 DataSet: For our Course Recommendation System, we use the Coursera dataset, which is available in Kaggle. The dataset contains a total of 3522 rows and 7 attributes. Each row in this dataset is defined by attributes such as Course Name, University, Course Description, Course Rating, Difficulty level, Course URL, and Skills.

5. Result Analysis

The final results of the customized recommendations are described in this section with screen visualizations. The dataset we collected includes the course name, the degree of difficulty, and the course ratings. The course description and associated skills are also included. Figure 2. shows the home page of the system with the imported dataset. The combination of Course Name, Course Rating, Difficulty Level, University, and Course Description constitutes the 'tags' attribute in this case. Figure 3. displays the list of recommendations for the courses to the individual users. Figure 4. shows the recommendations based on the keyword.

Figure 2. Home page of the Recommender System using Streamlit

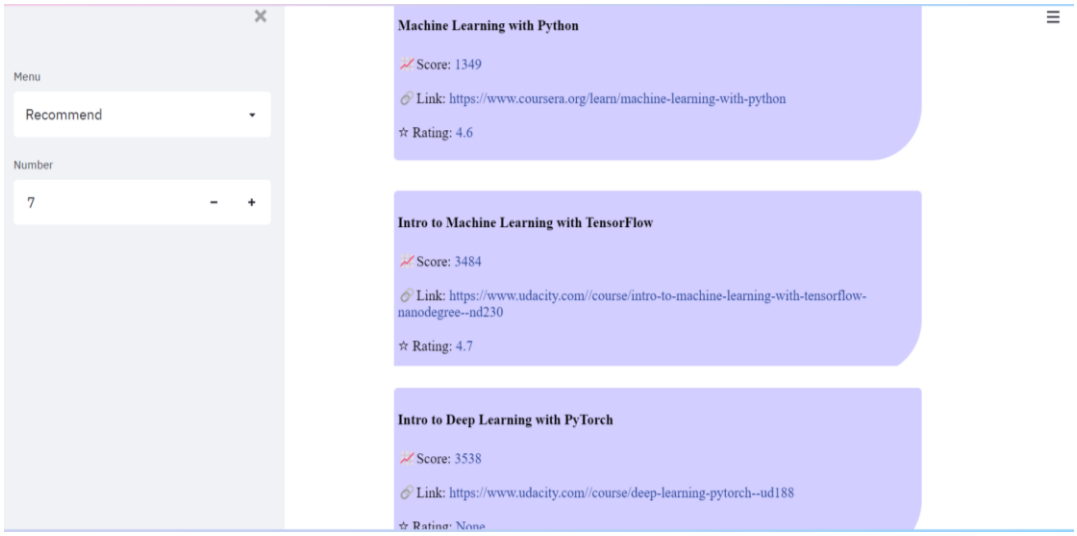

Figure 3. Recommend page recommending courses to the user

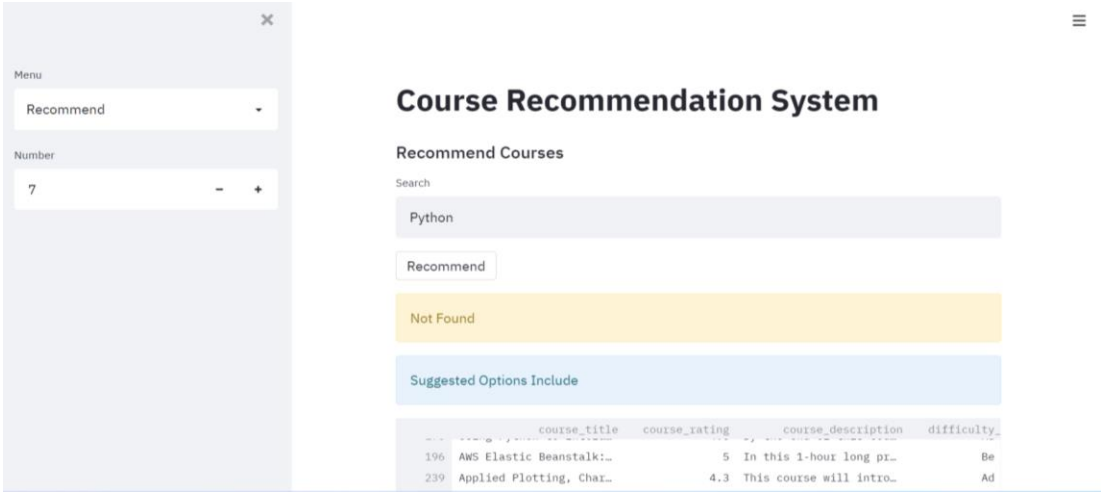

Figure 4. Recommending courses based on the keyword

6. Conclusion

Selecting online courses as per user interest from a large amount of available online courses is a challenging task. Hence there is a need of developing a personalized recommendation system, which can recommend relevant courses to the students as per their requirements. In this paper, we propose a system with Collaborative Filtering that can recommend the courses to the students. Our proposed system takes user interest and item similarity into consideration to recommend the courses. In the future, this work can be extended for recently advanced courses with user cluster-based interest, behavioral patterns, and also hidden pattern knowledge extraction.

References:

[1] Kashada, Abubaker, Awatif Isnoun, and Nureddin Aldali. "Effect of information overload on decision's quality, efficiency and time." *International Journal of Latest Engineering Research and Applications* 5, no. 1 (2020): 53-58.

[2] Fayyaz, Zeshan, Mahsa Ebrahimian, Dina Nawara, Ahmed Ibrahim, and Rasha Kashef. "Recommendation systems: Algorithms, challenges, metrics, and business opportunities." *applied sciences* 10, no. 21 (2020): 7748.

[3] Jain, Sarika, Anjali Grover, Praveen Singh Thakur, and Sourabh Kumar Choudhary. "Trends, problems and solutions of recommender system." In *International conference on computing, communication & automation*, pp. 955-958. IEEE, 2015.

[4] Aguilar, Jose, Priscila Valdiviezo-Díaz, and Guido Riofrio. "A general framework for intelligent recommender systems." *Applied computing and informatics* 13, no. 2 (2017): 147-160.

[5] Di Rocco, Juri, Davide Di Ruscio, Claudio Di Sipio, Phuong T. Nguyen, and Riccardo Rubei. "Development of recommendation systems for software engineering: the CROSSMINER experience." *Empirical Software Engineering* 26, no. 4 (2021): 69.

[6] Zhou, Kun, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. "Towards topic-guided conversational recommender system." *arXiv preprint arXiv:2010.04125* (2020).

[7] Aljunid, Mohammed Fadhel, and Manjaiah Dh. "An efficient deep learning approach for collaborative filtering recommender system." *Procedia Computer Science* 171 (2020): 829-836.

[8] Singh, Pramod, and Pramod Singh. "Machine learning deployment as a web service." *Deploy Machine Learning Models to Production: With Flask, Streamlit, Docker, and Kubernetes on Google Cloud Platform* (2021): 67-90.

[9] Ricci, Francesco, Lior Rokach, and Bracha Shapira. "Introduction to recommender systems handbook." In *Recommender systems handbook*, pp. 1-35. Boston, MA: springer US, 2010.

[10] Sarwar, Badrul, George Karypis, Joseph Konstan, and John Riedl. "Item-based collaborative filtering recommendation algorithms." In *Proceedings of the 10th international conference on World Wide Web*, pp. 285-295. 2001.

[11] Imran, Hazra, Mohammad Belghis-Zadeh, Ting-Wen Chang, Kinshuk, and Sabine Graf. "PLORS: a personalized learning object recommender system." *Vietnam Journal of Computer Science* 3 (2016): 3-13.

[12] Lynn, N. D., and A. W. R. Emanuel. "A review on Recommender Systems for course selection in higher education." In *IOP Conference Series: Materials Science and Engineering*, vol. 1098, no. 3, p. 032039. IOP Publishing, 2021.

[13] O'Mahony, Michael P., and Barry Smyth. "A recommender system for on-line course enrolment: an initial study." In *Proceedings of the 2007 ACM conference on Recommender systems*, pp. 133-136. 2007.

[14] Lu, Jinliang. "A Survey of Online Course Recommendation Techniques." *Open Journal of Applied Sciences* 12, no. 1 (2022): 134-154.

[15] Chang, Pei-Chann, Cheng-Hui Lin, and Meng-Hui Chen. "A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems." *Algorithms* 9, no. 3 (2016): 47.