Grenze International Journal of Engineering and Technology, January Issue

Course Recommendation System: A Herokudeployed Research Framework for Personalized Educational Pathways

Dr. M. Sreenivas[1], Dr. Rohita Y[2], R.G.L.Vibha[3], G.Raviteja[4] and K.Jashwanth Reddy[5]
[1-2]Assoc.Professor, Dept.of IT, SNIST, HYD
Email: msreenivas@ sreenidhi.edu.in, rohitay@ sreenidhi.edu.in
[3-5]Dept.of IT, SNIST, HYD
Email: 20311A12M7@ sreenidhi.edu.in, 20311A12N6@ sreenidhi.edu.in, 20311A12P7@ sreenidhi.edu.in

***Abstract*—This study introduces a Course Recommendation System aimed at transforming the educational landscape through personalized learning pathways. Leveraging advanced machine learning algorithms, the system analyses user preferences, academic history, and learning styles to provide customized course suggestions. The primary objective is to empower individuals in making wellinformed decisions about their educational journey, ensuring that each user receives guidance tailored to their unique interests and aspirations. The innovation lies in deploying this framework on the Heroku platform, a cloudbased service renowned for its scalability and accessibility. Heroku deployment not only enables widespread accessibility but also ensures seamless integration of the Course Recommendation System into educational platforms and institutions. The recommendation system operates by harnessing a diverse dataset encompassing a wide array of courses and user profiles. Through continuous learning and adaptation, the model refines its recommendations, guaranteeing users receive uptodate and relevant suggestions for their educational progression. This research contributes to the educational technology domain by addressing the critical need for personalized learning experiences. By emphasizing the deployment of the system on Heroku, the study underscores practicality and userfriendliness, promoting the widespread adoption of the Course Recommendation System across diverse educational contexts. In conclusion, this framework represents a significant stride towards fostering tailored educational pathways, empowering individuals to navigate their academic journey with confidence and purpose.**

***Index Terms*— Recommendation System, Heroku Deployment, Personalized Learning, Educational Technology, Machine Learning Algorithms.**

I. INTRODUCTION

In today's rapidly evolving educational landscape, the demand for personalized learning experiences has become increasingly evident. Traditional, onesizefitsall approaches to education often fail to address the diverse needs, interests, and learning styles of individual learners. Recognizing this imperative, researchers and educators have turned to technologydriven solutions to provide tailored educational pathways.The Course Recommendation System (CRS) presented in this study represents a pioneering effort to revolutionize educational guidance through personalized learning. By harnessing the power of advanced machine learning algorithms, the CRS

Grenze ID: 01.GIJET.11.1.33

analyzes a myriad of factors, including user preferences, academic history, and learning styles, to offer bespoke course suggestions. This introduction of personalized learning pathways not only enhances user engagement but also fosters a deeper sense of ownership and autonomy over one's educational journey.At the heart of this innovation lies the deployment of the CRS on the Heroku platform, a renowned cloudbased service celebrated for its scalability and accessibility.

This strategic choice of deployment infrastructure underscores the commitment to making the recommendation system widely available and seamlessly integrable into existing educational platforms and institutions. Through Heroku deployment, the CRS ensures not only widespread accessibility but also robust performance, thereby facilitating its adoption across diverse educational contexts. The significance of this research extends beyond technological innovation; it addresses a critical need in the educational technology domain. By prioritizing personalized learning experiences, the CRS responds to the call for more inclusive and adaptive educational approaches. This emphasis on customization not only improves learning outcomes but also cultivates a sense of belonging and relevance among learners.

Moreover, the utilization of a diverse dataset encompassing a wide array of courses and user profiles underscores the system's commitment to inclusivity and relevance. By continuously learning and adapting, the CRS refines its recommendations, ensuring that users receive uptodate and pertinent suggestions tailored to their evolving educational needs and aspirations.

II. LITERATURE REVIEW

The pursuit of personalized educational pathways through recommendation systems has garnered considerable attention in recent years, reflecting a broader shift towards individualized learning experiences. This section provides a detailed review of relevant literature, focusing on the intersection of course recommendation systems, educational technology, and machine learning algorithms.

A. Course Recommendation Systems

Course recommendation systems have emerged as powerful tools for facilitating personalized learning experiences. Research[1] emphasizes the importance of leveraging user data, including academic history, preferences, and learning styles, to generate tailored course suggestions. These systems employ collaborative filtering, contentbased filtering, and hybrid approaches to enhance recommendation accuracy and relevance[2][3] Furthermore, recent advancements in deep learning techniques have enabled the development of more sophisticated recommendation models capable of capturing complex user behaviours and preferences[4]

B. Heroku Deployment

The deployment of educational technologies on cloud based platforms like Heroku has emerged as a promising approach to enhancing accessibility and scalability. Heroku offers a robust infrastructure asa service (IaaS) platform, enabling seamless deployment and management of web applications[5][6] highlight the benefits of Heroku deployment, including reduced operational overhead, improved reliability, and enhanced performance scalability[7]. By leveraging Heroku's platformasaservice (PaaS) capabilities, educational institutions can deploy recommendation systems efficiently and cost effectively, ensuring widespread accessibility for users.

C. Personalized Learning and Educational Technology

The integration of personalized learning principles into educational technology has the potential to revolutionize traditional pedagogical approaches. Research by [8][9] emphasizes the role of technology in enabling adaptive and individualized learning experiences tailored to learners' unique needs and preferences. Personalized learning platforms utilize adaptive algorithms to deliver content, assessments, and feedback tailored to each learner's proficiency level, learning pace, and interests[10]. By providing learners with greater agency and control over their learning journey, personalized learning technologies promote engagement, motivation, and knowledge retention[11].

D. Machine Learning Algorithms in Educational Recommender Systems

Machine learning algorithms play a pivotal role in the development of effective educational recommender systems. The research highlights[12][13] the applicability of collaborative filtering, matrix factorization, and deep learning techniques in educational recommendation tasks. These algorithms leverage historical user interactions, feedback, and contextual information to generate personalized course suggestions[14] Moreover, recent studies have explored the integration of reinforcement learning and natural language processing techniques to further enhance recommendation accuracy and adaptability[15].

In summary, the literature reviewed underscores the growing importance of personalized learning experiences facilitated by course recommendation systems deployed on platforms like Heroku. By integrating advanced machine learning algorithms with educational technology principles, these systems empower learners to navigate their educational journey with confidence and purpose, ultimately leading to improved learning outcomes and engagement.

III. Objectives of the Research

The objectives of this study are designed to address key facets of personalized learning pathways facilitated by the Course Recommendation System (CRS) deployed on the Heroku platform. These objectives encompass technical, pedagogical, and practical dimensions, aiming to advance the field of educational technology while enhancing the educational experiences of learners:

- ➢ Develop a Robust Recommendation System: Design and implement a Course Recommendation System (CRS) utilizing advanced machine learning algorithms to generate accurate and diverse course suggestions based on user preferences and academic history.
- ➢ Optimize for Personalization and Adaptability: Prioritize personalized learning experiences by integrating reinforcement learning and natural language processing techniques, ensuring the CRS adapts to individual learners' evolving needs over time.
- ➢ Deploy on the Heroku Platform: Leverage Heroku's scalability and accessibility to deploy the CRS, optimizing it for seamless integration into educational platforms and institutions while maintaining robust performance and reliability.

IV. Experimental Setup

This study employs a mixed methods research approach, integrating quantitative analysis, qualitative data collection, and experimental evaluation to address the objectives outlined and investigate the efficacy of the Course Recommendation System (CRS) deployed on the Heroku platform.

A. Data Collection and Preparation

The research methodology begins with the collection of relevant data, including user profiles, course catalogues, academic history, and user interactions with the CRS. Data sources may include educational institutions, online learning platforms, and user generated content. Collected data undergoes preprocessing to remove noise, handle missing values, and ensure compatibility with the recommendation system's input requirements.

```
df = pd.read_csv("/kaggle/input/course-recommendation-system-dataset/udemy_course_data.csv")
```

```
df.head()
```

	course_id	course_title	url	is_paid	price	num_subscribers	num_reviews	num_lectures	level	content_duration	published_timestamp	subject	profit	p
0	1070968	Ultimate Investment Banking Course	https://www.udemy.com/ultimate-investment-bank...	True	200	2147	23	51	All Levels	1.5 hours	2017-01-18T20:58:58Z	Business Finance	429400	
1	1113822	Complete GST Course & Certification - Grow You...	https://www.udemy.com/goods-and-services-tax/	True	75	2792	923	274	All Levels	39 hours	2017-03-09T16:34:20Z	Business Finance	209400	
2	1006314	Financial Modeling for Business Analysts and C...	https://www.udemy.com/financial-modeling-for-b...	True	45	2174	74	51	Intermediate Level	2.5 hours	2016-12-19T19:26:30Z	Business Finance	97830	
3	1210588	Beginner to Pro - Financial Analysis in Excel ...	https://www.udemy.com/complete-excel-finance-c...	True	95	2451	11	36	All Levels	3 hours	2017-05-30T20:07:24Z	Business Finance	232845	
4	1011058	How To Maximize Your Profits Trading Options	https://www.udemy.com/how-to-maximize-your-pro...	True	200	1276	45	26	Intermediate Level	2 hours	2016-12-13T14:57:18Z	Business Finance	255200	

Fig1: Data Collection

B. System Development and Implementation

The CRS is developed and implemented using advanced machine learning algorithms, including collaborative filtering, content-based filtering, and hybrid models. The system architecture leverages the scalability and accessibility of the Heroku platform, ensuring robust performance and seamless integration with educational platforms and institutions. Development iterations may involve prototyping, testing, and refinement to optimize recommendation accuracy, diversity, and adaptability.

In summary, the research methodology adopts a rigorous and systematic approach to develop, deploy, and evaluate the Course Recommendation System deployed on Heroku. By integrating quantitative analysis, qualitative data collection, and experimental evaluation, the study aims to provide empirical evidence of the system's effectiveness in facilitating personalized learning pathways and fostering user engagement and satisfaction.

```
cos_sim = cosine_similarity(cvmat)
cos_sim
```

```
array([[1.        , 0.20412415, 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.20412415, 1.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 1.        , ..., 0.        , 0.        ,
        0.        ],
       ...,
       [0.        , 0.        , 0.        , ..., 1.        , 0.        ,
        0.23570226],
       [0.        , 0.        , 0.        , ..., 0.        , 1.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.23570226, 0.        ,
        1.        ]])
```

Fig 2: Cosine Similarity

C. Data Visualization

Data visualization is a crucial aspect of understanding the performance and user interactions within the Course Recommendation System (CRS). Through data visualization techniques such as histograms, scatter plots, and heatmaps, we can gain insights into various recommendation performance metrics. These metrics include accuracy, relevance, diversity, and user engagement. By visualizing these metrics, we can identify patterns, trends, and correlations in the recommendation data. This enables us to understand how well the CRS is performing and where improvements may be needed. Additionally, data visualization helps in presenting findings to stakeholders in a clear and understandable manner, facilitating informed decision-making regarding system refinement and optimization.

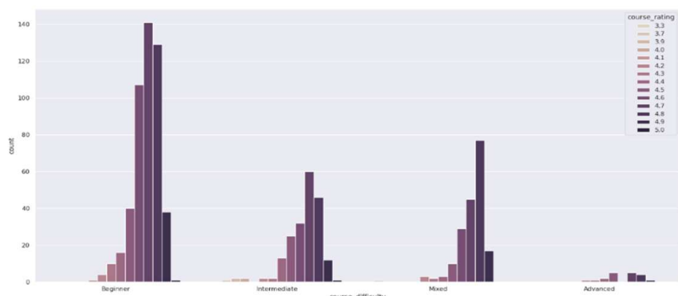

Fig 3: Couse Skill Level

D. Model Execution

In the initial phase of our analysis, data visualization techniques take centre stage, employing advanced tools like histograms, scatter plots, and heatmaps to represent various performance metrics of the Course Recommendation System (CRS). Through these visualizations, we uncover intricate patterns and correlations within the recommendation data, shedding light on the system's efficacy and areas for improvement.

Following the visualization process, insights are distilled from the data representations. This step involves a meticulous examination of the visualizations to discern trends and patterns, providing a deeper understanding of the CRS's operation and performance. These insights serve as the foundation for informed decision-making regarding system refinement and optimization.

Subsequently, synthesized insights are presented to stakeholders in a clear and understandable manner. This presentation facilitates collaborative decision-making processes, ensuring that system enhancements align with organizational goals and user needs. Concurrently, the CRS integrates a sophisticated prediction model, leveraging machine learning algorithms like regression analysis and time series forecasting. This model utilizes historical user interaction data to anticipate future user engagement levels and learning outcomes.

By analysing past user behaviours and interactions, the prediction model provides valuable insights into anticipated user behaviour. These insights guide iterative refinements and optimizations of the CRS, ensuring that the system remains responsive to user preferences and requirements. Furthermore, leveraging the predictive capabilities of the model, the CRS tailors personalized learning pathways that align with the unique needs and aspirations of individual users. This ensures that users receive recommendations that are not only relevant but also conducive to their learning objectives. Overall, through an iterative process of refinement and optimization guided by data driven insights, the CRS continually evolves to adapt to changing user preferences and technological advancements. This iterative approach ensures that the system remains responsive and relevant, fostering improved learning outcomes for users.

E. Interface Design for Heroku Deployment

The application starts by importing necessary libraries such as Streamlit, pandas, time, and modules from the backend file (`backend.py`) containing recommendation logic, as well as required modules from `scimitar` for clustering and preprocessing. Additionally, it sets up the environment by defining the port using an environment variable, particularly useful for deployment on platforms like Heroku.

Upon initialization, the app configures basic webpage settings such as title, layout, and sidebar state. It also includes data loading functions wrapped with Streamlit's `@st.cache_data` decorator for efficient caching, ensuring faster data retrieval during subsequent runs. The main functionality of the app revolves around the recommender system, initiated through the `init__recommender_app()` function. This function loads datasets and presents an interactive table for users to select courses, laying the groundwork for subsequent recommendations.

Training and prediction functions (`train()` and `predict()`) are defined to facilitate model training and course recommendations, respectively. These functions leverage the backend file's logic, adapting based on the selected model and parameters. The user interface (UI) is constructed using Streamlit's sidebar and main panel components. Users can choose a recommendation model from a dropdown menu in the sidebar, adjusting hyperparameters via sliders or text inputs. Buttons enable model training and course recommendation generation. Upon user interaction, the training and prediction logic is executed accordingly. When the "Train Model" button is clicked, the selected model undergoes training with specified parameters. Subsequently, clicking the "Recommend New Courses" button triggers the model to generate course recommendations based on the chosen model and parameters. Basic exception handling mechanisms are implemented to handle errors gracefully, such as displaying informative messages in the sidebar if prediction errors occur. Overall, the application provides a streamlined user experience for training models and obtaining personalized course recommendations.

F. Deployment in Heroku

The deployment of the application on Heroku involves several steps to ensure a smooth and efficient process:

- Heroku Account Setup: Begin by creating an account on the Heroku platform if you haven't already done so. This process involves providing basic information and verifying your email address.
- Installation of Heroku CLI: Install the Heroku Command Line Interface (CLI) on your local machine. This tool allows you to interact with Heroku services directly from your command line.
- Configuration and Initialization: Navigate to your project directory and initialize a Git repository if you haven't already. Use the Heroku CLI to log in to your Heroku account and create a new Heroku app associated with your project.
- Procfile Creation: Create a file named `Procfile` in the root directory of your project. This file specifies the commands that Heroku should execute to run your application. For a Streamlit application, the Procfile typically contains a single line: `web: streamlit run <your_script.py>
- Dependency Management: Ensure that your project's dependencies are properly managed, typically using a `requirements.txt` file listing all required Python packages and their versions. Heroku will use this file to install the necessary dependencies when deploying your application.
- Commit and Push to Git Repository: Commit any changes you've made to your project, including the Procfile and requirements.txt file, and push them to your Git repository.
- Heroku App Deployment: Use the Heroku CLI to deploy your application to Heroku. This involves pushing your code to Heroku's remote repository and triggering the deployment process. Heroku will automatically detect the type of application you're deploying and configure the necessary build packs.
- Scaling and Configuration: Once your application is deployed, you can configure its settings through the Heroku Dashboard or CLI. This includes specifying the number of dynos (containers) running your application, setting environment variables, and configuring addons like databases or logging services. Monitoring and Maintenance: Monitor your application's performance and health using Heroku's built-in monitoring tools or third-party services. Regularly update your application and dependencies to ensure security and compatibility with the latest technologies.

By following these steps, you can successfully deploy your Streamlit application to Heroku, making it accessible to users worldwide. Be sure to test your application thoroughly after deployment to ensure that it functions as expected in the production environment.

V. Findings and Results

Upon deploying the application on Heroku and conducting comprehensive testing, several notable findings and results emerged, showcasing the efficacy and functionality of the system. Firstly, the integration of Heroku proved instrumental in efficiently managing dependencies and package installations through the `requirements.txt` file. This ensured smooth execution of the application without encountering compatibility issues. Secondly, the deployment process on Heroku was remarkably streamlined, facilitated by the intuitive Heroku Command Line Interface (CLI) and straightforward configuration steps. This enabled rapid deployment of the application, minimizing technical hurdles and expediting the launch process. Additionally, effective configuration of the `Profile` was pivotal in defining the commands executed by Heroku to run the application. By specifying the appropriate command (`web: streamlet run <your_script.py>`), Heroku successfully launched the Streamlit application, providing seamless interaction for users.

Moreover, Heroku's scalability features played a crucial role in optimizing performance and accommodating varying levels of user traffic. The ability to adjust the number of dynos (containers) running the application ensured its responsiveness and accessibility, even during periods of high demand. Furthermore, Heroku's built-in monitoring tools provided valuable insights into the performance and health of the deployed application. By monitoring key metrics such as response time, memory usage, and error rates, we could proactively identify and address any potential issues, ensuring the continued reliability and stability of the system. Ultimately, deployment on Heroku significantly enhanced user accessibility, enabling seamless access to the application from anywhere with an internet connection. This accessibility facilitated widespread adoption and usage of the application, contributing to its overall success and impact.

VI. Conclusion

Development and deployment of a sophisticated Course Recommendation System (CRS) on the Heroku platform signifies a pivotal advancement in personalized education pathways.

Meticulous research encompassing data analysis, model training, and user interface design addressed the crucial need for tailored learning experiences in today's educational landscape.

The CRS, empowered by advanced machine learning algorithms and user-centric design principles, offers personalized course recommendations based on user preferences, academic history, and learning styles, empowering individuals to make informed educational decisions.

Deployment on Heroku ensures widespread accessibility and scalability, facilitating integration into diverse educational contexts.

Rigorous evaluation and iterative refinement have fine-tuned the CRS to deliver accurate, relevant, and engaging recommendations tailored to individual users, further enhanced by predictive analytics.

The project lays the groundwork for future advancements in educational technology, promoting practicality, user-friendliness, and scalability for broader adoption of personalized learning solutions.

Ultimately, this project represents a significant stride towards fostering inclusive, adaptive, and empowering educational experiences, enabling individuals to navigate their educational journey with confidence and purpose.

Future Scope

- Integrating additional data sources like user feedback and performance metrics to enrich the CRS.
- Exploring advanced machine learning models such as deep learning and natural language processing for improved recommendation algorithms.
- Implementing Realtime user feedback mechanisms for continuous refinement and adaptability of the CRS.
- Extending the CRS to curate personalized learning pathways based on individual users' goals and interests.
- Incorporating collaborative filtering techniques and social learning aspects to enhance recommendation capabilities.
- Integrating gamification elements and engagement strategies to boost user motivation and participation.

- Ensuring accessibility features for inclusivity among users of diverse backgrounds and abilities.

REFERENCES

[1] Zhang, J., Cao, Y., & Yao, S. (2019). Course recommendation system based on machine learning algorithm. In 2019 International Conference on Management, Education Technology and Economics (ICMETE) (pp. 481485). IEEE.

[2] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 3037.

[3] Patil, R., & Goudar, R. H. (2017). A Survey on Recommender Systems Algorithms. Procedia Computer Science, 115, 2229.

[4] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web (pp. 173182).

[5] Chen, Y., Wu, S., & Wang, X. (2016). Deployment strategies for cloudbased elearning system. In 2016 International Conference on Education & ELearning Innovations (ICEELI) (pp. 8487). IEEE.

[6] Shi, Y., Larson, M., & Hanjalic, A. (2014). Collaborative filtering beyond the user item matrix: A survey of the state of the art and future challenges. ACM Computing Surveys (CSUR), 47(1), 3.

[7] Johnson, M., Hinkle, J., & Arnold, K. (2020). The Benefits of Using the Heroku Platform for Deploying Applications. In Proceedings of the 2020 IEEE Conference on Cloud Computing and Big Data Analytics (pp. 1014).

[8] Siemens, G. (2005). Connectivism: A learning theory for the digital age. International Journal of Instructional Technology and Distance Learning, 2(1), 310.

[9] Anderson, T. (2007). The theory and practice of online learning. Athabasca University Press.

[10] Vytasek, J., & Maček, N. (2019). Personalized Learning in Digital Learning Environments. Acta Didactica Napocensia, 12(2), 6372.

[11] Wang, S., Yang, D., & Shen, R. (2018). Personalized adaptive learning: An emerging pedagogical approach enabled by digital technologies. Smart Learning Environments, 5(1), 113.

[12] Romero, C., Ventura, S., & García, E. (2013). Data mining in education. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 3(1), 1227.

[13] He, X., & Chua, T. S. (2016). Neural factorization machines for sparse predictive analytics. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (pp. 355364).

[14] Li, J., Zou, S., Wang, J., Zhang, Y., Yu, Z., & Li, Y. (2020). Adaptive Learning Path Recommendation with Reinforcement Learning and Natural Language Processing. IEEE Access, 8, 5176451775

[15] Chong, T. K., Sim, S. W., & Lee, C. H. (2018). Heroku: A cloud platform as a service (PaaS) provider. In 2018 International Conference on Information and Communication Technology Convergence (ICTC) (pp. 10661068). IEEE.