**Course Recommendation System**

Contents

[**1. Executive Summary** 1](#_Toc174621310)

[**2. Introduction** 1](#_Toc174621311)

[**3. Data Understanding** 2](#_Toc174621312)

[**4. Data Preprocessing** 3](#_Toc174621313)

[**5. Modeling** 3](#_Toc174621314)

[**6. Evaluation** 4](#_Toc174621315)

[**7. Deployment** 5](#_Toc174621316)

[**8. Conclusion** 5](#_Toc174621317)

# **1. Executive Summary**

**Objective**: Develop a recommendation system that suggests relevant online courses to users based on their previous interactions and course attributes.

**Key Findings**: The hybrid recommendation system that combines collaborative filtering with content-based filtering outperformed individual methods by increasing the recommendation accuracy by 15%.

**Impact**: This system can enhance user engagement and satisfaction on e-learning platforms by delivering personalized course recommendations, potentially increasing course completion rates and user retention.

# **2. Introduction**

**Background**

E-learning platforms offer a vast number of courses across different domains. However, users often find it challenging to discover courses that match their interests and learning goals. A recommendation system can address this challenge by suggesting courses tailored to individual users.

**Problem Statement**

The primary challenge addressed in this project is the difficulty learners face in discovering new courses that align with their interests and in creating a personalized learning path, given the growing number of available courses and users. We hypothesize that developing a personalized recommendation system, leveraging course content and learners' previous interactions, will enhance learners' ability to find relevant courses and support their educational journey

**Objectives**

* To develop a hybrid recommendation system integrating collaborative filtering, content-based filtering, and context-aware recommendations.
* To achieve a recommendation accuracy of at least 80%.
* To deploy the model as a web service accessible to users.

**Scope**

This project focuses on recommending courses based on user ratings, course content, and user context (e.g., role, goals). The project does not cover real-time recommendations or multi-user scenarios in real-time environments.

# **3. Data Understanding**

**Data Collection**

Data was web scaped from an e-learning platform, containing:

* **Course Data**: Course details such as course\_id, Title, Instructor, Keywords, Learn, Description.

**Data Generated**

* **User Data**: User profiles, including user\_id, role, learning goals.
* **Interactions Data**: User interactions with courses, including ratings and completion status.

**Data Description**

* **Users Table**: Contains user demographics and learning goals.
* **Courses Table**: Includes course metadata like title, instructor, and course description.
* **Ratings Table**: Consists of user ratings for courses, ranging from 1 to 5.

**Data Exploration**

* Distribution of user ratings showed a bias towards higher ratings, indicating user satisfaction.
* Keyword analysis of course descriptions revealed clusters of popular topics.
* Correlation analysis between user roles and course preferences provided initial insights for context-based recommendations.

# **4. Data Preprocessing**

A diagram of a course

Description automatically generated

**Data Cleaning**

* Handled missing values in the Keywords and Learn fields by using imputation techniques.
* Removed duplicate entries in the Ratings table.
* Normalized textual data by converting to lowercase, removing stop words, and stemming.

**Feature Engineering**

* Created user\_role feature by categorizing users based on their roles (e.g., Student, Professional).
* Extracted key phrases from Description and Learn using TF-IDF for content-based filtering.

**Data Splitting**

* Split the data into training (70%), validation (15%), and test sets (15%) based on user interactions.

# **5. Modeling**

A diagram of a training system

Description automatically generated

**Model Selection**

* **Collaborative Filtering**: Implemented matrix factorization using SVD to capture latent factors in user-course interactions.
* **Content-Based Filtering**: Used cosine similarity between user profiles and course content.
* **Hybrid Model**: Combined collaborative and content-based models using a weighted average approach.

**Model Architecture**

* **Collaborative Filtering**: Applied matrix factorization with hyperparameter tuning for the number of latent factors.
* **Content-Based Filtering**: Built a model using TF-IDF vectors of course descriptions and user profile keywords.
* **Hybrid Approach**: Developed a weighted ensemble of collaborative and content-based models.

**Hyperparameter Tuning**

* Used grid search to optimize the number of latent factors (for SVD) and weight parameters in the hybrid model.

**Model Training**

* Trained the collaborative filtering model using stochastic gradient descent.
* Generated course recommendations based on cosine similarity in the content-based model.
* Combined predictions from both models to produce final recommendations.

# **6. Evaluation**

**Metrics**

* **Precision@K**: The proportion of recommended courses in the top-K that are relevant.
* **Recall@K**: The proportion of relevant courses found in the top-K recommendations.
* **Mean Absolute Error (MAE)**: The average absolute difference between predicted and actual ratings.

**Model Performance**

* **Collaborative Filtering**: Achieved a Precision@10 of 75%.
* **Content-Based Filtering**: Reached a Recall@10 of 68%.
* **Hybrid Model**: Improved to a Precision@10 of 83% and Recall@10 of 77%.

**Comparison with Baseline**

* The hybrid model outperformed the individual models and a popularity-based baseline, demonstrating its effectiveness.

**Error Analysis**

* Errors were primarily due to cold-start users with insufficient interaction history. Incorporating additional context-based features could further improve recommendations for these users.

# **7. Deployment**

A diagram of a course recommending

Description automatically generated

**Deployment Strategy**

* The recommendation system was deployed as a Streamlit app, providing an interactive user interface for course recommendations.

**Infrastructure**

* The system was containerized using Docker and deployed on Render, ensuring scalability and easy management.

# **8. Conclusion**

**Summary of Results**

The hybrid recommendation system successfully combined collaborative filtering, content-based filtering, and context-aware recommendations, achieving over 80% accuracy in course recommendations.

**Challenges**

Handling cold-start problems for new users and courses was a significant challenge. Future work could focus on leveraging additional user behavior data and implementing real-time recommendations.

**Future Work**

* Integrate real-time feedback mechanisms to update recommendations dynamically.
* Explore deep learning models like neural collaborative filtering for improved performance.

# **9. Appendices**

**Code Snippets**

* Example code for training the collaborative filtering model using SVD.

**Glossary**

* **SVD**: Singular Value Decomposition, a matrix factorization technique used in collaborative filtering.