Personalized Learning Recommender

Solving the Paradox of Choice in Online Education with AI & Deep Learning

The Problem: Choice Overload

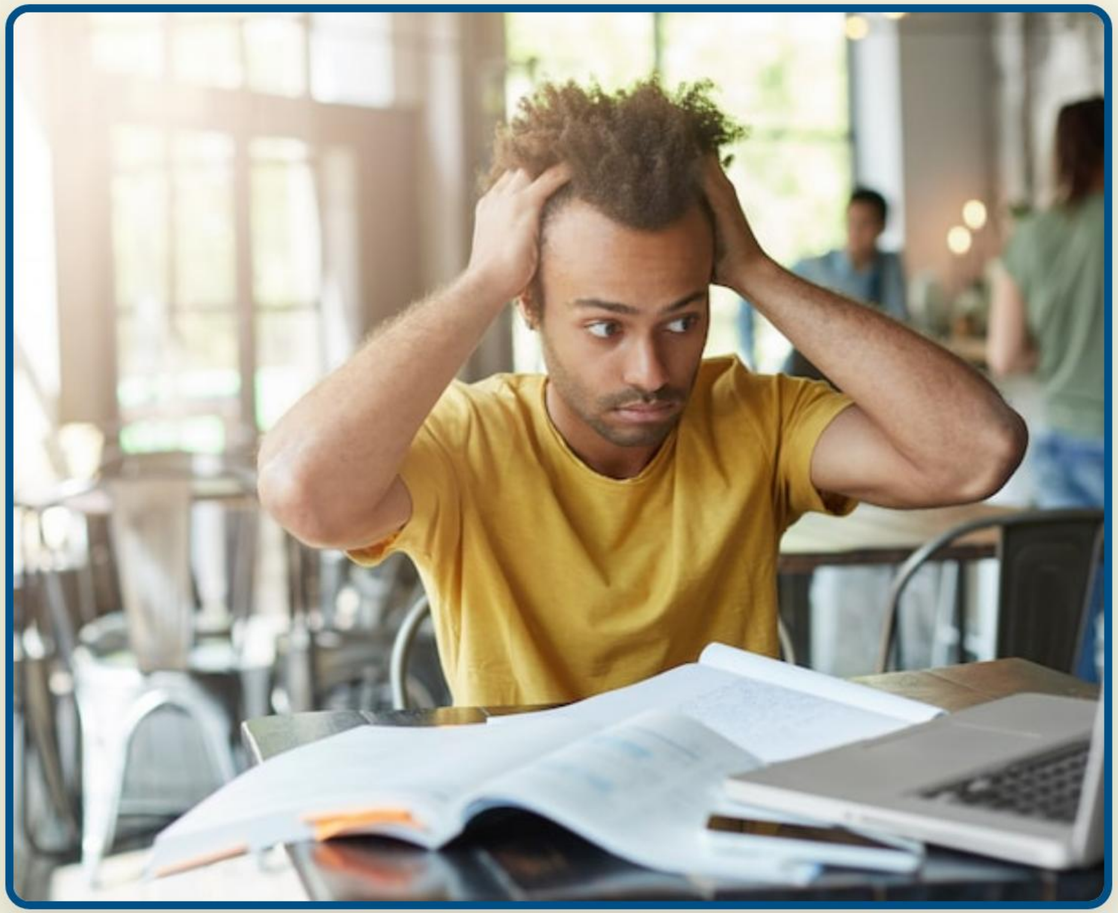
Analysis Paralysis

In the expansive world of online education, learners are often overwhelmed. Platforms host thousands of overlapping courses, creating a paradox of choice.

Disconnected Experience

Traditional "Top 10" lists fail to account for unique learning histories. A novice needs different guidance than a data scientist, yet platforms often treat them the same.

Result: High drop-off rates and wasted time.



System Architecture

End-to-End Workflow

The system bridges user interaction with complex backend logic using a Streamlit-based MVC architecture.

- **Interaction Layer:** Captures user intent and history via a responsive UI.
- **Logic Controller:** Routes requests to the appropriate mathematical model.
- **Computation Core:** Handles matrix operations, embeddings, and distance calculations.

Data Flow

User Input → Streamlit UI

↓

Backend Controller (Router)

↓

Model Engine (Neural Network / KNN)

↓

Interactive Frontend Features

Course Selection

Built with AgGrid, allowing users to efficiently search and select from thousands of processed courses to build their initial profile.

Dynamic Tuning

Real-time hyperparameter adjustments. Users can tweak 'k' neighbors, epochs, or cluster counts and see immediate results.

Model Switching

Instant switching between 9 different algorithms, enabling direct comparison of recommendation strategies.

The Algorithmic Core

Traditional Methods

Content-Based: Matches course descriptions and genres to user profiles.

Clustering (K-Means + PCA): Groups users into cohorts based on behavior. If you are in "Cluster A", you see what others in "Cluster A" liked.

Collaborative (KNN & NMF): Finds mathematical similarities in the user-item rating matrix.

Deep Learning

Neural Network (RecommenderNet): A custom TensorFlow/Keras model that learns low-dimensional embeddings for users and items.

Hybrid Models: Extracts these learned embeddings and feeds them into Random Forest classifiers or Linear Regressors for refined predictions.

Deep Learning Approach

At the heart of our advanced recommendations lies **RecommenderNet**.

Instead of simple keyword matching, this model learns 16-dimensional vector representations (embeddings) for every user and course.

By training on user interaction history, the model places similar courses close together in this vector space, allowing us to compute affinity scores via dot products.

Neural Network Architecture

Logic Flow

1. **Input:** User ID and Item ID pairs.
2. **Embedding Layers:** Look up dense vectors.
3. **Dot Product:** Calculates similarity.
4. **Bias Addition:** Adjusts for user optimism.
5. **Activation:** ReLU ensures positive ratings.

Hybrid Classification

Combining Powers

We don't stop at the Neural Network. We extract the **learned embeddings** and treat them as high-quality features.

These features are then fed into a **Random Forest Classifier**. This hybrid approach leverages the feature extraction power of Deep Learning with the robust decision-making capabilities of Ensemble Tree models.

Technology Stack

- **Python 3.x:** Core Logic
- **Streamlit:** Frontend UI
- **Pandas & NumPy:** Data Engineering
- **TensorFlow (Keras):** Deep Learning
- **Scikit-Learn:** KNN, PCA, Random Forest
- **Surprise Lib:** Matrix Factorization

User Journey Demo

1

History

User selects previously taken courses (e.g., "Intro to Python").

2

Config

Selects "Neural Network" model and sets epochs to 10.

3

Training

System trains model & generates embeddings in real-time.

4

Predict

System ranks unseen courses by computed affinity scores.

Future Improvements

Infrastructure

Database Integration: Migrate from CSV to PostgreSQL to handle concurrent users and data integrity.

Async Processing: Use Celery for background model training to keep the UI responsive.

Logic Enhancements

Cold-Start Logic: Average item embeddings of selected courses instead of generic user averages.

Explainability (XAI): Visualize *why* a course was recommended (e.g., "Because you liked Course A...").

Questions?

Thank you for your attention.

Image Sources

https://img.freepik.com/free-photo/international-afro-american-student-feeling-stressed-keeping-hands-his-head-staring-laptop-screen-frustration-despair_273609-7488.jpg?semt=ais_hybrid&w=740&q=80

Source: www.freepik.com

https://images.stockcake.com/public/8/5/b/85bcbe14-4d6f-4954-82b4-00a52b967a65_large/neural-network-visualization-stockcake.jpg

Source: stockcake.com

https://towardsdatascience.com/wp-content/uploads/2025/06/t2.jpg

Source: towardsdatascience.com