Context-Aware Course Recommendation System

## 1. Problem Statement

Online learning platforms host thousands of courses across multiple domains. Traditional recommendation systems often rely solely on course similarity or user preferences, ignoring the real-life learning context such as:

* Available study time per week
* Device used for learning (mobile vs. desktop)
* Study schedule (morning vs. night learning)

Without considering context, recommendations may not be practical. For instance, suggesting long technical courses to a user with limited time or using a mobile device may not be effective.

**Objective:**

Design a *Context-Aware Course Recommendation System* that integrates:

* Course content similarity
* Course popularity
* Learner context (time availability, device, study hours)

The system employs a hybrid recommendation approach with an interactive Streamlit interface.

## 2. Dataset Details

**Dataset Used:** Coursera.csv

| Column Name | Description |
| --- | --- |
| partner | Course provider (e.g., Google, IBM) |
| course | Course title/name |
| skills | Skills taught |
| rating | User rating (0–5 scale) |
| reviewcount | Number of reviews |
| level | Difficulty level (Beginner/Intermediate) |
| certificatetype | Available certification option |
| duration | Course duration description |
| crediteligibility | Academic credit availability |

**Derived Fields:**

| Feature | Purpose |
| --- | --- |
| skills\_list | Tokenized list of skills |
| reviewcount\_num | Numeric cleaned review counts |
| duration\_months | Duration converted into months |
| text\_blob | Combined text fields for TF-IDF encoding |

This dataset supports content-based filtering, popularity analysis, and context-aware recommendations.

## 3. Methodology

The pipeline follows a hybrid recommendation architecture:

### Step 1: Data Cleaning & Preprocessing

* Handle missing values
* Normalize rating and duration
* Convert categorical features into usable formats

### Step 2: Feature Engineering

* **TF-IDF Vectorization:** Transform skills into numeric vectors
* **Scaling:** Normalize numeric features such as ratings, review count, and duration

### Step 3: Context Modeling

* **User Inputs:**
  + Hours per week
  + Device (mobile/desktop)
  + Preferred study timing (currently a placeholder)
* **Context Scoring:**
  + Short courses recommended for limited weekly hours
  + Desktop users suggested longer programs

### Step 4: Hybrid Score Calculation

* **Similarity Score:** Based on TF-IDF cosine similarity
* **Popularity Score:** Normalized rating + log(reviewcount)
* **Context Score:** Fit with study constraints

**Final Score Formula:** [ = 0.45() + 0.20() + 0.35() ]

### Step 5: Streamlit UI

* **Sidebar Inputs:** Time availability, device, study preferences, filters
* **Tabs:**
  + Recommendations (cards + table view)
  + EDA visualizations (charts and insights)

## 4. Results

Since no explicit ground truth exists, **qualitative evaluation** was performed:

| Context Input | Recommendation Behavior |
| --- | --- |
| Limited hours/week | Short, quick courses prioritized |
| Mobile users | Flexible, short-term skill courses recommended |
| Desktop users | Long, structured specialization programs |
| Preferred study time | Neutral (missing dataset info) |

**Performance Characteristics:**

| Aspect | Result |
| --- | --- |
| Cold-start handling | Effective via content-based TF-IDF |
| Personalization | Moderate (context-aware adjustments) |
| Interpretability | High — scoring logic is transparent |
| Scalability | Efficient for large course datasets |

### Summary

The Context-Aware Course Recommendation System is:

* **Explainable:** Transparent scoring and ranking
* **Adaptive:** Adjusts recommendations based on context
* **Scalable:** Lightweight and efficient

**Limitations:** - No real-time feedback loop - Limited semantic understanding (TF-IDF) - Placeholder feature for preferred study timing

**Future Enhancements:** - Reinforcement learning using user click behavior - Deep contextual embeddings (e.g., BERT) for better content understanding - Temporal personalization from usage logs - Cross-domain recommendations

**Technologies Used:** Python, Streamlit, Pandas, NumPy, Scikit-learn, Altair

**Conclusion:** This project demonstrates a practical, explainable, and lightweight Context-Aware Recommender System. By integrating contextual cues, it personalizes online learning recommendations effectively. Future work can enhance personalization and semantic understanding through feedback-driven models and deep learning approaches.