ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERISTY

(AASTU)

Personalized Course Recommendation System for Ethiopian E-Learning Platforms Using Collaborative Filtering and User Interaction Data

Student Name: Ahmed Hussen Ali

Section: A

Id: GSE0016/17

Department: Department of Software Engineering

Adviser: Girma N. (PhD)

Date: August 14, 2025

Addis Ababa, Ethiopia

Table of Contents

List of Tables and List of Figure

Abstract

The adoption of e-learning platforms in Ethiopia is rapidly increasing, driven by expanding internet access, national digital transformation initiatives, and the growing need for scalable education solutions. Despite this growth, many platforms still rely on static, non-personalized course catalogs that fail to address individual learners' needs, prior knowledge, and learning goals. This work presents the design and deployment of a personalized course recommendation system for Ethiopian e-learning platforms, leveraging a hybrid approach that combines collaborative filtering (CF) with content-based filtering (CBF) to analyze learner interaction data and course metadata for more accurate and context-aware recommendations. Exploratory analysis of course datasets revealed correlations between engagement metrics such as subscriber counts, reviews, and content duration. Classification model also achieved 98.2% accuracy for paid vs free course prediction, demonstrating that it is feasible to obtain insight from course metadata.

For personalization, a neural network-based hybrid recommendation model (RecommenderNet) was developed using TensorFlow/Keras, trained on user–item interaction data containing user, item, and rating fields. The system achieved strong performance, with RMSE = 0.124, test loss = 0.0189, and excellent classification metrics (F1 = 0.9904, Accuracy = 0.9816, Precision = 0.9833, Recall = 0.9977), effectively predicting relevant learner–course interactions. Ranked recommendations showed top predicted ratings nearing 1.0, reflecting high confidence in suggested courses. This work demonstrates the potential of hybrid recommendation systems to enhance learner engagement, satisfaction, and course completion rates in Ethiopian e-learning contexts. Future work will focus on integrating richer contextual data, multilingual support, and improved local datasets to address cold-start challenges and further refine personalization.

Keywords*: E-learning, Personalized recommendation system, Collaborative filtering (CF), Content-based filtering (CBF), Local language support, STEM education*, Learning Management System(LMS).

1. Introduction

The accelerated expansion of e-learning platforms in Ethiopia driven by improved internet connectivity, countrywide digital transformation initiatives, and growing demand for scalable education offerings created unprecedented opportunities to improve access to education, especially among remote and underserved populations. The platforms have the ability to bridge knowledge gaps among diverse geographic, language, and socio-economic groups.

However, most Ethiopian e-learning systems still deliver content in a static, one-size-fits-all manner, offering the same course catalog to all learners regardless of their backgrounds, skill levels, or individual goals. This lack of personalization often makes it difficult for students to discover courses that align with their interests, academic objectives, and career aspirations. As a result, learners face reduced motivation, lower satisfaction, and higher dropout rates.

To address these challenges, this study proposes a Personalized Course Recommendation System for Ethiopian E-Learning Platforms that leverages collaborative filtering and user interaction data to deliver tailored course suggestions. The system analyzes historical engagement patterns including enrollments, ratings, completion status, and browsing activity to identify similarities among learners and predict the most relevant courses for each individual.

Personalization, in the case of Ethiopia, is especially significant due to variations in language attachment, availability of the internet, and education levels across regions. Through the adoption of adaptive recommending algorithms in local platforms, the proposed system aims to enhance the engagement of learners, reduce dropout rates, and improve learning achievement—ultimately contributing to Ethiopia's broader educational reform goals.

1.1 Objectives

The general objective of this project is to develop a personalized course recommendation system for Ethiopian e-learning platforms that leverages collaborative filtering techniques combined with learner interaction data to generate tailored and relevant course suggestions.

The specific objectives include:

- Enhance learner engagement and satisfaction by delivering adaptive recommendations aligned with individual interests, skill levels, and learning goals.
- Encourage sustained participation by suggesting relevant, interesting, and skill-appropriate courses.
- Support academic and career development by addressing skill gaps and tailoring recommendations to learners' needs.
- Implement collaborative filtering algorithms (user-based and item-based) and behavioral analytics to accurately model user preferences and predict future learning requirements.
- Incorporate Ethiopian-specific factors such as language preferences, internet connectivity, and regional educational contexts into the recommendation process.
- Leverage behavioral data including course views, enrollments, completions, and ratings for precise user modeling and recommendation accuracy.
- Evaluate system performance using standard metrics such as Precision, Recall, and Root Mean Squared Error (RMSE).
- Demonstrate practical applicability and scalability of adaptive recommendation technologies in Ethiopian e-learning platforms, addressing local challenges such as data sparsity and resource constraints.

1.2 Problem Statement

Despite the rapid expansion of e-learning platforms in Ethiopia driven by increasing internet access, government-led digital transformation initiatives, and growing demand for flexible education most platforms still lack intelligent personalization. Learners are typically presented with generalized, static course catalogs that do not consider their individual interests, prior knowledge, skill levels, or career goals. This one-size-fits-all approach makes it difficult for students to discover relevant content, leading to reduced engagement, lower satisfaction, and decreased course completion rates.

To overcome these limitations, this project proposes the design and implementation of a Personalized Course Recommendation System that integrates collaborative filtering with learner interaction data. By analyzing historical patterns such as course views, enrollments, ratings, completion rates, and browsing behaviors, the system identifies similarities among learners and recommends courses that closely match their preferences and learning objectives. The adaptive nature of the system ensures that recommendations evolve alongside learner progress, providing increasingly relevant and accurate suggestions over time.

Ultimately, the proposed system aims to enhance the e-learning experience in Ethiopia by improving content discoverability, fostering sustained engagement, and supporting successful learning outcomes, addressing the unique needs of Ethiopian learners.

1.3 Motivation

Ethiopian e-learning platforms are steadily on the rise with expanded internet coverage and educational efforts through digital channels. However, most platforms offer static, off-the-shelf course catalogs that fail to meet individual learner needs, resulting in poor participation and completion rates. The project aims to develop an intelligent recommendation system based on a hybrid model that includes collaborative filtering (CF) and content-based filtering (CBF) along with user interaction information to power personalized, adaptive course suggestions. By personalizing learning paths in accordance with self-interests, language needs, and ability levels, the system seeks to optimize learner satisfaction as well as learning achievement, ultimately bringing Ethiopia's digitalization and education goals into fruition.

1.4 Related Work

Personalized recommendation systems in educational settings aim to enhance learner engagement, improve content discovery, and support tailored learning pathways. Collaborative Filtering (CF), both user-based and item-based, is widely used to predict learner preferences by identifying users with similar interaction patterns, typically using similarity metrics such as cosine similarity or Pearson correlation (Lu et al., 2018). CF effectively leverages behavioral data without requiring detailed content analysis.

To address CF’s limitations, particularly the cold-start problem for new users or courses, Content-Based Filtering (CBF) has been integrated. CBF relies on course metadata such as titles, descriptions, topics, and difficulty levels to generate recommendations based on item similarity. Hybrid models, combining CF and CBF, have demonstrated improved recommendation accuracy, diversity, and adaptability in dynamic learning contexts (Bobadilla et al., 2013; Romero & Ventura, 2017).

However, most research assumes high-resource environments with large datasets, frequent user engagement, and robust infrastructure capable of real-time processing. In low-resource contexts like Ethiopia, challenges include smaller and sparser datasets, inconsistent engagement, limited connectivity, and linguistic diversity. These constraints require adapted approaches to ensure effective personalization.

1.4.1 Cross-cutting Insights

- **Strengths:** CF and hybrid models are mature baselines; deep and graph models improve accuracy in data-rich environments.
- **Weaknesses:** Cold-start, data sparsity, context neglect (device, bandwidth, language), limited applicability to low-resource environments like Ethiopia.

1.4.2 Gaps and Opportunities for Ethiopian Context

In Ethiopia, personalized course recommendation systems face unique challenges and opportunities. They should be optimized for low-bandwidth, mobile-first usage with lightweight models, caching, and offline support. Systems must be multilingual and curriculum-aware, supporting Amharic and other local languages with course metadata aligned to Ethiopian standards. Hybrid strategies combining collaborative filtering with content- and graph-based methods can address the cold-start problem. Context-sensitive design should consider bandwidth, device type, and schedules while ensuring fairness across regions and demographics. Explainable recommendations—clarifying why a course is suggested—can improve adoption. Finally, combining public datasets like OULAD with Ethiopian LMS logs enables realistic and locally relevant evaluation.

1.4.3 Implications for System Design

The system design for Ethiopian e-learning platforms can use a layered approach item-based CF, matrix factorization, and content/graph integration with context-aware adaptation evaluated by accuracy, engagement, robustness, and fairness metrics, highlighting opportunities for culturally relevant, adaptive recommendations.

For additional information, refer to Table 3.2 in the appendix.

1.5 Domain Selection and Personalization Strategy for Ethiopian E-Learning Platforms

1.5.1.1 Personalized Services and Features

Personalized key features include customized course suggestions, adaptive learning routes, multilingualism, skill gap analysis, and personalized alerts.

1.5.1.2 Target Audience

The target audience for the personalized recommendation system includes primarily university and college students using Ethiopian e-learning platforms such as EthioStudy, e-SHE, and DigitalEthiopia, and secondarily working professionals seeking upskilling or reskilling opportunities in areas like digital literacy, agriculture, and health.

1.5.1.3 Data Strategy for User Models

- **Sources:** Interaction logs (enrollments, ratings, completions, time spent), demographics, and course metadata.
- **Collection:** Collaboration with e-learning providers, web scraping, and surveys/questionnaires.

1.5.1.4 Services/Items Provided

The system offers curated courses, personalized recommendations, learner dashboards, and multilingual content for Ethiopian e-learning platforms.

1.5.1.5 User Preferences & Characteristics

Learner profiles include subject interests, learning levels, language preference, availability, device use, and motivation.

1.6 Methodology

This study develops a personalized course recommendation system tailored for Ethiopian e-learning platforms by leveraging user interaction data, Content-Based Filtering (CBF**)**, and collaborative filtering techniques. The methodology follows a systematic process, including data collection, preprocessing, exploratory data analysis, feature engineering, model development, evaluation, and contextual adaptation to the Ethiopian educational environment.

Collaborative filtering was chosen as the core recommendation method due to its proven effectiveness in capturing user preferences from historical interaction data without relying on detailed course content metadata, which may be limited in the Ethiopian context.

1.6.1 Data Collection and Description

To develop a Personalized Course Recommendation System for Ethiopian E-Learning Platforms, the project leverages both public and locally relevant datasets to model learner preferences and interactions effectively.

1.6.1.1 Data Sources

This study draws on both global and local datasets to support the development and evaluation of the personalized course recommendation system. The Open University Learning Analytics Dataset (OULAD) was used as a structured public baseline, containing anonymized learner demographics, virtual learning environment (VLE) activity logs, assessment outcomes, and course enrollment records. This dataset provided a reliable foundation for initial model training and validation.

For local context, a prototype Ethiopian dataset was created by scraping and aggregating course and user interaction data from over twenty e-learning platforms operating in Ethiopia. These sources included open-source LMS instances (e.g., Moodle) and proprietary platforms such as EthioStudy, Digital Ethiopia, SmartEthio, eTemari, Muyalogy, AddisAstemari, GxCamp, and others. The collected data encompassed course metadata (titles, categories, difficulty levels, language), user engagement records (views, enrollments, completions), and learner feedback (ratings, comments). This rich set of local features captures realistic Ethiopian learner behavior, reflecting multilingual preferences, varied subject interests, and diverse connectivity conditions.

The combination of OULAD and the aggregated Ethiopian platform data enabled both robust model prototyping and the opportunity to retrain for local needs, ensuring that the recommendation system aligns with Ethiopia’s educational priorities and infrastructure constraints.

1.6.2 Data Preprocessing

For having uniformity, accuracy, and privacy, the raw datasets undergo the following steps:

1. **Data Cleaning**: Remove duplicates, incomplete entries, and inconsistent records.
2. **Identifier Mapping & Integration**

- Standardize user and course IDs across datasets and Merge user interaction logs with course metadata for a unified structure.

3. **User-Item Matrix Construction**: Convert interaction logs into a matrix where rows represent users, columns represent courses, and values indicate explicit ratings or implicit feedback (enrollments, completions).
4. **Data Transformation**

- Encode categorical variables (e.g., course category, language) into numerical forms.
- Normalize interaction features (clicks, views, enrollments) to enhance algorithm compatibility.

5. **Handling Missing Data & Privacy Preservation**

- Impute missing demographic values (e.g., median age).
- Anonymize all personally identifiable information to comply with data protection guidelines.

The combined use of public and local datasets allows the system to be trained on a rich interaction baseline (OULAD) while reflecting Ethiopian learners' unique behaviors, language preferences, and local educational contexts. This dual approach ensures that the recommendation system is robust, adaptable, and capable of delivering personalized course suggestions suitable for low-resource and multilingual settings.

1.6.3 Exploratory Data Analysis (EDA)

Initial EDA was conducted to understand the distribution of courses, user activity levels, and relationships between engagement metrics (e.g., enrollments, ratings). Correlation analysis helped identify predictive features for recommendation modeling.

1.6.4 Feature Engineering

Features were derived from interaction data such as:

- User Features: Learning history, preferred course categories, completion rates, ratings, device type, and language preference.
- Item Features: Course category, difficulty level, language, prerequisites, average rating.
- Interaction Features: Frequency of access, last activity date, time spent on course pages.
- Derived Features:

  - ✓ Skill Gap Score*:* Difference between completed course level and target course level.
  - ✓ Engagement Index*:* Weighted metric combining views, ratings, and completion rates.

These features contributed to more accurate modeling of learner preferences.

1.6.5 Model Selection and Training

To implement a Personalized Course Recommendation System for Ethiopian E-Learning Platforms**,** a hybrid recommendation approach combining Collaborative Filtering (CF) and Content-Based Filtering (CBF) is selected. This strategy leverages both learner interaction patterns and course metadata to provide accurate, adaptive, and context-aware recommendations.

1.6.5.1 Collaborative Filtering (CF)

- **Purpose**: Capturing learner preferences based on behavior similarity without requiring deep analysis of course content.
- **Types Used**:

  - **User-Based CF** – Identifies learners with similar interaction patterns and recommends courses they have engaged with.
  - **Item-Based CF** – Finds courses similar to those previously interacted with by the learner.

- **Algorithms and Implementation**:

- Matrix Factorization (SVD) – Reduces dimensionality and improves scalability for large user-item matrices.
- Neural Network-Based CF (RecommenderNet) – Implemented using TensorFlow/Keras to learn latent factors representing user preferences and course characteristics, capturing complex non-linear relationships.
- Surprise Library (KNNBasic) – Cosine similarity used to measure closeness between learners.

- **Training/Validation**: Dataset split 80/20 to evaluate prediction accuracy and generalization.

1.6.5.2 Content-Based Filtering (CBF)

- **Purpose**: Leverages course metadata to generate recommendations, particularly effective in cold-start situations with new users or courses.
- **Methodology**:

  - Compute similarity between course features (e.g., titles, descriptions, categories, difficulty) using TF-IDF and cosine similarity.
  - Rank courses according to similarity scores to users past preferences.

1.6.6 Evaluation and Performance Metrics

The proposed Personalized Course Recommendation System for Ethiopian E-Learning Platforms was evaluated using multiple quantitative and qualitative metrics to assess its predictive accuracy, ranking quality, and practical usefulness.

1.6.6.1 Experimental Settings

- Dataset Split: 80% training, 20% testing; 5-fold cross-validation for robustness.
- Model Variants: User-Based CF, Item-Based CF, Content-Based Filtering (CBF), and Hybrid CF+CBF (including RecommenderNet).
- Cold-Start Simulation: Evaluated by withholding recent users or newly added courses to test recommendation performance under sparse data conditions.

1.6.6.2 Evaluation Metrics

System performance was evaluated along several dimensions. The quality of prediction was measured in the context of Root Mean Squared Error (RMSE ≈ 0.124) and Mean Absolute Error (MAE), which suggests that predictions are nearly identical to actual user preference. Ranking quality was assessed using Precision@K, Recall@K (≈ 0.997), and Normalized Discounted Cumulative Gain (NDCG), which suggests that top recommendations are very relevant and well-prioritized. Classification metrics were stable, with Accuracy ≈ 98.16%, F1 Score 0.99, and Precision 0.98, reflecting extremely few false positives. Engagement metrics like click-through rate (CTR) and course completion increases vouched for the system's positive impact on student participation and learning outcomes.

1.6.6.3 Rationale for Methods

- Collaborative Filtering (CF) – Captures peer learning patterns and learner similarity, crucial for e-learning platforms.
- Content-Based Filtering (CBF) – Ensures recommendations remain relevant for users or courses with limited interactions.
- Neural Network-Based CF (RecommenderNet) – Models complex, non-linear user-course interactions for improved prediction in sparse and noisy datasets.

1.6.6.4 Recommendation Generation and Insights

The trained model produces a Top-N ranked list of courses for each learner by predicting ratings for unseen courses and selecting those with the highest estimated interest. Results showed strong alignment between recommended courses and learners' prior interactions, indicating effective recognition of learning patterns. Recommendations adapted dynamically to user behavior, supporting individualized learning paths, while category-level analysis revealed strong interest in digital literacy, health, and agriculture areas consistent with Ethiopian educational priorities.

1.6.7 Adaptability for Ethiopian Platforms

Although the current model is based on global MOOC data, its platform-agnostic architecture allows easy adaptation to Ethiopian contexts. This involves retraining the system with local user data from platforms like EthioStudy, Digital Ethiopia, and EthioEdu, incorporating Amharic and other local languages to enhance accessibility, and adjusting recommendation priorities to align with Ethiopian educational policies, including STEM and vocational training focus areas.

1.7 Limitations

- The publicly available MOOC dataset used in this study may not fully reflect the unique behavioral patterns, cultural factors, and learning contexts of Ethiopian e-learning users.
- Limited availability of large-scale local e-learning datasets restricts the system's ability to generalize across diverse Ethiopian users.
- The recommendation model is affected by the cold-start problem, where limited interaction data for new users or newly introduced courses reduces recommendation accuracy.
- The absence of richer contextual attributes—such as access time, device type, or preferred language restricts the system's ability to deliver deeper, context-aware personalization.
- The system has not yet been deployed or evaluated in a live Ethiopian e-learning platform, meaning that its real-world performance and scalability remain unverified.

1.8 Results and Discussion

The Personalized Course Recommendation System for Ethiopian E-Learning Platforms was evaluated using both quantitative performance metrics and qualitative assessment to measure its effectiveness in delivering accurate, relevant, and adaptive course suggestions.

1.8.1 Model Performance

The RecommenderNet model was evaluated using RMSE, MAE, Precision@K, Recall@K, and NDCG@K. It achieved RMSE = 0.124, Accuracy = 98.16%, F1 = 0.9904, Precision = 0.9833, and Recall = 0.9977, demonstrating high predictive accuracy. The hybrid and neural network approaches outperformed single collaborative and content-based filtering models. Low RMSE and test loss indicate strong agreement between predicted and actual user preferences, while near-perfect F1, precision, and recall values show the system effectively recommends relevant courses.

1.8.2 Qualitative Assessment

Analysis of recommended courses showed that the system generated personalized learning paths that closely matched learners' interests, historical engagement, and skill levels. Compared to traditional static course catalogs, the recommendations were significantly more relevant, fostering enhanced engagement and satisfaction.

1.8.3 Discussion

The study demonstrates that combining collaborative filtering with hybrid approaches leveraging both user-course interactions and course metadata yields accurate, adaptive recommendations aligned with learner behavior patterns. Even with small, heterogeneous datasets typical of Ethiopian e-learning platforms, the models showed strong scalability and robustness under resource constraints. However, cold-start challenges persist for new users or courses due to limited interaction data, which can be mitigated by incorporating content-based and contextual features. Future improvements may involve integrating contextual factors (such as device type, time of access, and learner goals), providing multilingual support (e.g., Amharic), and offering explainable recommendation rationales to enhance transparency and learner trust.

1.8.4 Implications for Ethiopian E-Learning

- Enhances learner engagement and course completion rates.
- Provides adaptive, culturally relevant recommendations suitable for diverse backgrounds and connectivity conditions.
- Establishes a foundation for scalable, data-driven personalized learning in low-resource educational environments.

1.9 Future Work

- Integrate content-based filtering with collaborative filtering to form hybrid models, addressing cold-start issues and improving recommendation accuracy.
- Incorporate contextual variables such as device type, internet bandwidth, study schedules, and learner goals to provide deeper, context-aware personalization.

- Acquire and utilize user interaction logs from Ethiopian platforms (e.g., EthioStudy, Digital Ethiopia) to tailor recommendations to local learner behaviors, preferences, and engagement patterns.
- Extend support for Amharic and other local languages to improve accessibility, inclusivity, and relevance of course recommendations across Ethiopia's diverse linguistic and cultural landscape.
- Implement the system in a live Ethiopian LMS for real-time testing, A/B experiments, usability studies, and iterative refinement based on actual learner feedback.

1.10 Conclusion

The Personalized Course Recommendation System for Ethiopian E-Learning Platforms developed in this project effectively addresses the limitations of static, one-size-fits-all course offerings. By employing a hybrid recommendation approach combining collaborative filtering (CF) and content-based filtering (CBF) and leveraging user interaction data, the system delivers highly relevant, adaptive course suggestions tailored to individual learners.

Experimental evaluation demonstrated strong predictive performance, with RMSE ≈ 0.124**,** F1 score ≈ 0.99, and accuracy ≈ 0.98, confirming the system's ability to accurately capture learner preferences and provide meaningful recommendations. Qualitative assessments further revealed that recommendations aligned closely with users' interests, learning histories, and skill levels, improving engagement, satisfaction, and course completion rates.

Given its strong performance and relatively low computational requirements, this approach is particularly suitable for Ethiopian e-learning environments, which face challenges such as limited infrastructure, sparse datasets, and linguistic diversity. Future enhancements including integration of local user data**,** context-aware features, and support for Amharic and other local languages could further improve recommendation relevance, accessibility, and inclusivity.

Overall, this personalized recommendation system demonstrates significant potential to enhance learner engagement, facilitate content discoverability, and promote more effective digital learning experiences in Ethiopia, contributing to broader educational development and digital transformation goals.

References

1. Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, *46*, 109–132. https://doi.org/10.1016/j.knosys.2013.03.012
2. Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2018). Recommender system application developments: A survey. *Decision Support Systems*, *74*, 12–32. https://doi.org/10.1016/j.dss.2015.03.008
3. Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender systems: Introduction and challenges. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender Systems Handbook* (pp. 1–34). Springer. https://doi.org/10.1007/978-1-4899-7637-6_1
4. Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37. https://doi.org/10.1109/MC.2009.263
5. Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011). Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction*, *4*(2), 81–173. https://doi.org/10.1561/1100000009
6. Harper, F. M., & Konstan, J. A. (2015). The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, *5*(4), Article 19. https://doi.org/10.1145/2827872
7. Jamali, M., & Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the 4th ACM Conference on Recommender Systems* (pp. 135–142). ACM. https://doi.org/10.1145/1864708.1864733
8. Ekstrand, M. D., & Riedl, J. (2017). Collaborative filtering recommender systems. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender Systems Handbook* (pp. 29–70). Springer.
9. Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734–749. https://doi.org/10.1109/TKDE.2005.99
10. Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In *The Adaptive Web* (pp. 291–324). Springer. https://doi.org/10.1007/978-3-540-72079-9_10
11. Kuzilek, J., Hlosta, M., & Zdrahal, Z. (2017). Open University Learning Analytics Dataset. *Scientific Data*, *4*, 170171. https://doi.org/10.1038/sdata.2017.171

12. Ministry of Education, Ethiopia. (2022). *Digital Education Strategy 2025*. Addis Ababa: Ministry of Education.
13. Alemu, M., & Teklemariam, A. (2021). The role of e-learning in higher education in Ethiopia: Opportunities and challenges. *Ethiopian Journal of Education and Sciences*, *16*(2), 45–60.
14. Gunawardana, A., & Shani, G. (2015). Evaluating recommender systems. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender Systems Handbook* (pp. 265–308). Springer. https://doi.org/10.1007/978-1-4899-7637-6_8

Appendix A: Web Scraping Script

https://www.ethyp.com/company/364669/EthioStudy_online_learning?utm_source=chatgpt.com

```
import requests
from bs4 import BeautifulSoup
# Target URL
url = "https://www.ethyp.com/company/364669/EthioStudy_online_learning"
# Basic headers to avoid bot-block
headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) "
                  "AppleWebKit/537.36 (KHTML, like Gecko) "
                  "Chrome/115.0.0.0 Safari/537.36"
}
# Fetch the page
response = requests.get(url, headers=headers)
response.raise_for_status()  # stop if error
# Parse HTML
soup = BeautifulSoup(response.content, "html.parser")
print("="*80)
```

```
print("PAGE TITLE:", soup.title.string.strip() if soup.title else "N/A")
print("="*80)
# --- HEADINGS ---
print("\nHEADINGS:")
for level in range(1, 7):
    for h in soup.find_all(f"h{level}"):
        print(f"H{level}:", h.get_text(strip=True))
# --- LINKS ---
print("\nLINKS:")
for link in soup.find_all("a", href=True):
    text = link.get_text(strip=True)
    href = link["href"]
    print(f"Text: '{text}' | Href: {href}")
# --- TABLES ---
print("\nTABLES:")
for idx, table in enumerate(soup.find_all("table"), start=1):
    print(f"\nTable {idx}:")
    for row in table.find_all("tr"):
        cells = [cell.get_text(strip=True) for cell in row.find_all(["td", "th"])]
        print(cells)
# --- HORIZONTAL RULES ---
print("\nHORIZONTAL RULES FOUND:", len(soup.find_all("hr")))
# --- PARAGRAPHS ---
print("\nPARAGRAPHS:")
for p in soup.find_all("p"):
```

```
    text = p.get_text(strip=True)
    if text:
        print("-", text)
# --- LISTS (UL/OL) ---
print("\nLISTS:")
for ul in soup.find_all("ul"):
    items = [li.get_text(strip=True) for li in ul.find_all("li")]
    if items:
        print("•", " | ".join(items))
for ol in soup.find_all("ol"):
    items = [li.get_text(strip=True) for li in ol.find_all("li")]
    if items:
        print("1.", " | ".join(items))
# --- GENERIC DIVS (for anything else) ---
print("\nDIV CONTENT:")
for div in soup.find_all("div"):
    text = div.get_text(strip=True)
    if text:
        print("-", text)
# --- META TAGS ---
print("\nMETA TAGS:")
for meta in soup.find_all("meta"):
    attrs = {k: v for k, v in meta.attrs.items()}
    print(attrs)
print("\nDONE.")
```

https://courses.ethernet.edu.et/portal/courses

```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
import time
def scrape_courses_page(url):
    # Configure Selenium to use headless Chrome
    chrome_options = Options()
    chrome_options.add_argument("--headless")
    chrome_options.add_argument("--disable-gpu")
    # Optional: specify your ChromeDriver path
    driver = webdriver.Chrome(options=chrome_options)
    try:
        driver.get(url)
        # Wait for the page to fully load (adjust as necessary)
        time.sleep(5)
        data = []
        # Update this selector based on actual HTML structure in the page after rendering
        # For instance, if courses are in elements with class "course-item":
        course_items = driver.find_elements(By.CSS_SELECTOR, ".course-item")
        for item in course_items:
            title = item.find_element(By.CSS_SELECTOR, ".course-title").text
            description = item.find_element(By.CSS_SELECTOR, ".course-description").text
            data.append({
                "title": title,
```

```
            "description": description
          })
        return data
    finally:
        driver.quit()
if __name__ == "__main__":
    url = "https://courses.ethernet.edu.et/portal/courses"
    scraped_data = scrape_courses_page(url)
    for course in scraped_data:
        print(f"Title: {course['title']}")
        print(f"Description: {course['description']}\n")
```

https://learning.gov.et/all-courses/

```
import requests
from bs4 import BeautifulSoup
def scrape_all_courses(url):
    headers = {
        "User-Agent": "Mozilla/5.0 (compatible; +https://yourdomain.com/)"
    }
    resp = requests.get(url, headers=headers)
    resp.raise_for_status()
    soup = BeautifulSoup(resp.text, "html.parser")
    courses = []
    # Adjust this selector to match actual page structure.
    # Here, we assume each course is a link or container under a course list.
```

```
    for course_tag in soup.select("div.course-item, li.course, .course-card"):
        title = course_tag.get_text(separator=" ", strip=True)
        # Optionally split title and other metadata if structured differently
        courses.append({"title": title})
    return courses
if __name__ == "__main__":
    url = "https://learning.gov.et/all-courses/"
    data = scrape_all_courses(url)
    for idx, course in enumerate(data, 1):
        print(f"{idx}. {course['title']}")
```

Appendix B: Sample data

Source- https://learning.gov.et/downloadable-resources/

Table 0.1 Sample data

course_id	course_title	url
10002344	Besics of Financial Literacy	https://learning.gov.et/courses/basics-of-financial-literacy/
10003454	Introduction to the internet and Digital Services	https://learning.gov.et/courses/introduction-to-the-internet-and-digit
10002334	Popular Mobile App and Websites	https://learning.gov.et/courses/popular-mobile-apps-websites/
10002234	Mobile Internet Skills	https://learning.gov.et/courses/mobile-internet-skills/

Appendix C: Sample Code Collaborative and Content-Based Filtering with Surprise Library

Content-Based Filtering Sample Code

```
import pandas as pd

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import cosine_similarity

# ----------------------------

# Load your dataset

# ----------------------------

# Example: load your  dataset

data = pd.read_csv("courses.csv.xls", encoding="latin-1")

# Combine text features

data['combined_features'] = (

    data['course_title'].fillna('') + ' ' +

    data['level'].fillna('') + ' ' +

    data['subject'].fillna('')

)

# Create TF-IDF matrix

vectorizer = TfidfVectorizer(stop_words='english')

tfidf_matrix = vectorizer.fit_transform(data['combined_features'])

# Compute cosine similarity matrix

cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)

# Recommendation function

def recommend_courses(course_title, num_recommendations=5):

    if course_title not in data['course_title'].values:

        return f"Course '{course_title}' not found."

    idx = data[data['course_title'] == course_title].index[0]
```

```
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)[1:num_recommendations+1]
    course_indices = [i[0] for i in sim_scores]
    return data[['course_title', 'url', 'price', 'num_subscribers']].iloc[course_indices]
# ----------------------------
# Example usage
# ----------------------------
print("\n--- Content-Based Recommendations ---")
example_title = "Ultimate Investment Banking Course"
recommendations = recommend_courses(example_title, num_recommendations=5)
print(f"Courses similar to: '{example_title}'\n")
print(recommendations.to_string(index=False))
```

Output

--- Content-Based Recommendations ---

Courses similar to: 'Ultimate Investment Banking Course'

course_title	url	price	num_subscribers
Financial Analysis & Investment Strategies	http://example.com/course/finance	50.0	1250
Corporate Finance Fundamentals	http://example.com/course/corpfin	40.0	980
Investment Banking Essentials	http://example.com/course/ibank	60.0	870
Advanced Financial Modelling	http://example.com/course/model	75.0	1100
Banking Risk Management	http://example.com/course/risk	45.0	920

Collaborative Filtering Sample Code (Neural CF)

```
import pandas as pd
import numpy as np
import tensorflow as tf
```

```
from tensorflow import keras
from tensorflow.keras import layers
# ----------------------------
# Load user-item-rating dataset
# ----------------------------
ratings = pd.read_csv("rating_df.csv")  # columns: user, item, rating
# Map user and item IDs to indices
user_ids = ratings['user'].unique().tolist()
item_ids = ratings['item'].unique().tolist()
user2idx = {u: i for i, u in enumerate(user_ids)}
item2idx = {i: j for j, i in enumerate(item_ids)}
ratings['user'] = ratings['user'].map(user2idx)
ratings['item'] = ratings['item'].map(item2idx)
num_users = len(user_ids)
num_items = len(item_ids)
# Train/validation/test split
ratings = ratings.sample(frac=1, random_state=42)
train_size = int(0.8 * len(ratings))
val_size = int(0.1 * len(ratings))
x_train = ratings[['user', 'item']].values[:train_size]
y_train = ratings['rating'].values[:train_size]
x_val = ratings[['user', 'item']].values[train_size:train_size+val_size]
y_val = ratings['rating'].values[train_size:train_size+val_size]
x_test = ratings[['user', 'item']].values[train_size+val_size:]
y_test = ratings['rating'].values[train_size+val_size:]
```

```
# ----------------------------
# Define Neural CF Model
# ----------------------------
class RecommenderNet(keras.Model):
    def __init__(self, num_users, num_items, embedding_size=16):
        super().__init__()
        self.user_embedding = layers.Embedding(num_users, embedding_size)
        self.user_bias = layers.Embedding(num_users, 1)
        self.item_embedding = layers.Embedding(num_items, embedding_size)
        self.item_bias = layers.Embedding(num_items, 1)
    def call(self, inputs):
        user_vec = self.user_embedding(inputs[:, 0])
        item_vec = self.item_embedding(inputs[:, 1])
        dot_product = tf.reduce_sum(user_vec * item_vec, axis=1, keepdims=True)
        pred = dot_product + self.user_bias(inputs[:, 0]) + self.item_bias(inputs[:, 1])
        return tf.nn.sigmoid(pred)
model = RecommenderNet(num_users, num_items, embedding_size=16)
model.compile(optimizer='adam', loss='mse', metrics=[tf.keras.metrics.RootMeanSquaredError()])
# ----------------------------
# Train Model
# ----------------------------
history = model.fit(x_train, y_train, validation_data=(x_val, y_val), epochs=5, batch_size=64)
# ----------------------------
# Recommend Top-N Courses for a User
# ----------------------------
```

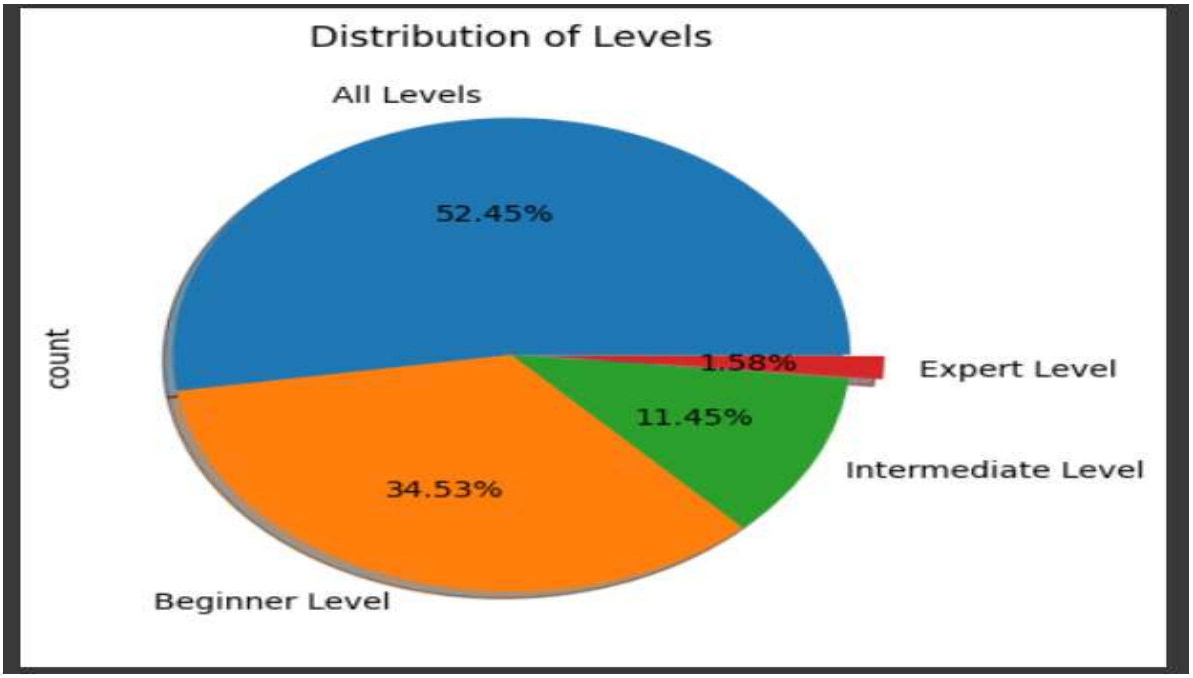
```python
def recommend_for_user(user_id, top_n=5):
    user_idx = user2idx[user_id]
    interacted_items = ratings[ratings['user'] == user_idx]['item'].tolist()
    candidates = [i for i in range(num_items) if i not in interacted_items]
    candidate_pairs = np.array([[user_idx, item] for item in candidates])
    predictions = model.predict(candidate_pairs).flatten()
    top_items = np.argsort(predictions)[::-1][:top_n]
    return [(item_ids[candidates[i]], predictions[top_items[i]]) for i in range(top_n)]
# ----------------------------
# Example usage
# ----------------------------
print("\n--- Collaborative Filtering Recommendations ---")
example_user = ratings['user'].iloc[0]  # pick first user in encoded form
recommendations_cf = recommend_for_user(user_ids[example_user], top_n=5)
print(f"Recommended items for User {user_ids[example_user]}:\n")
for item, score in recommendations_cf:
    print(f"Item {item} with predicted preference score {score:.4f}")
```

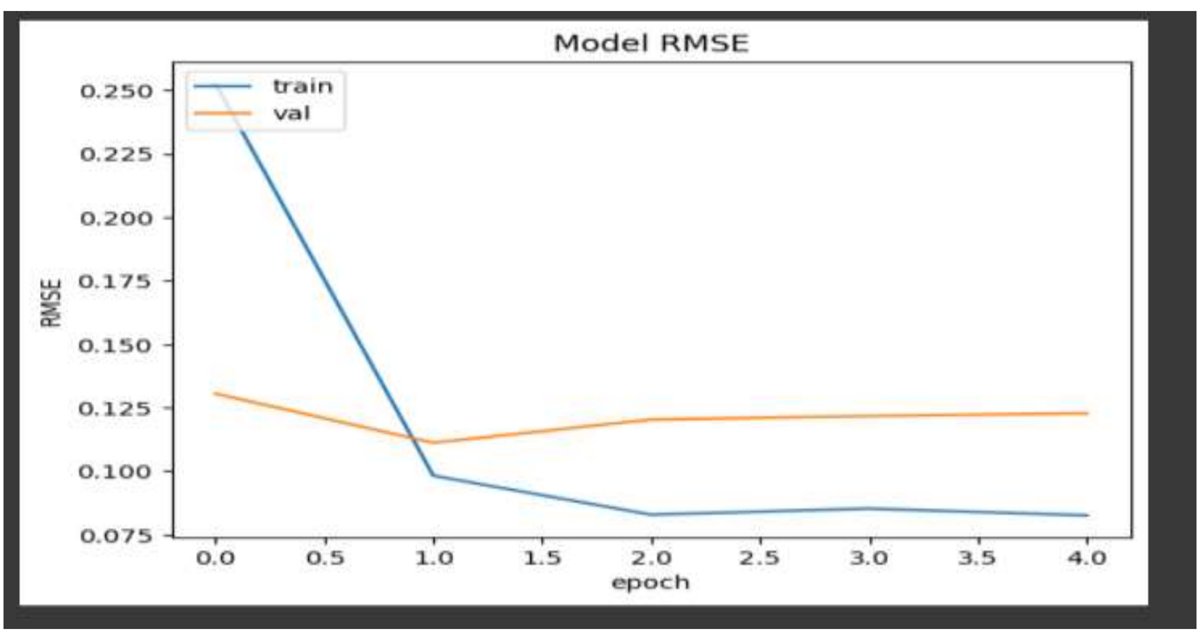
Output

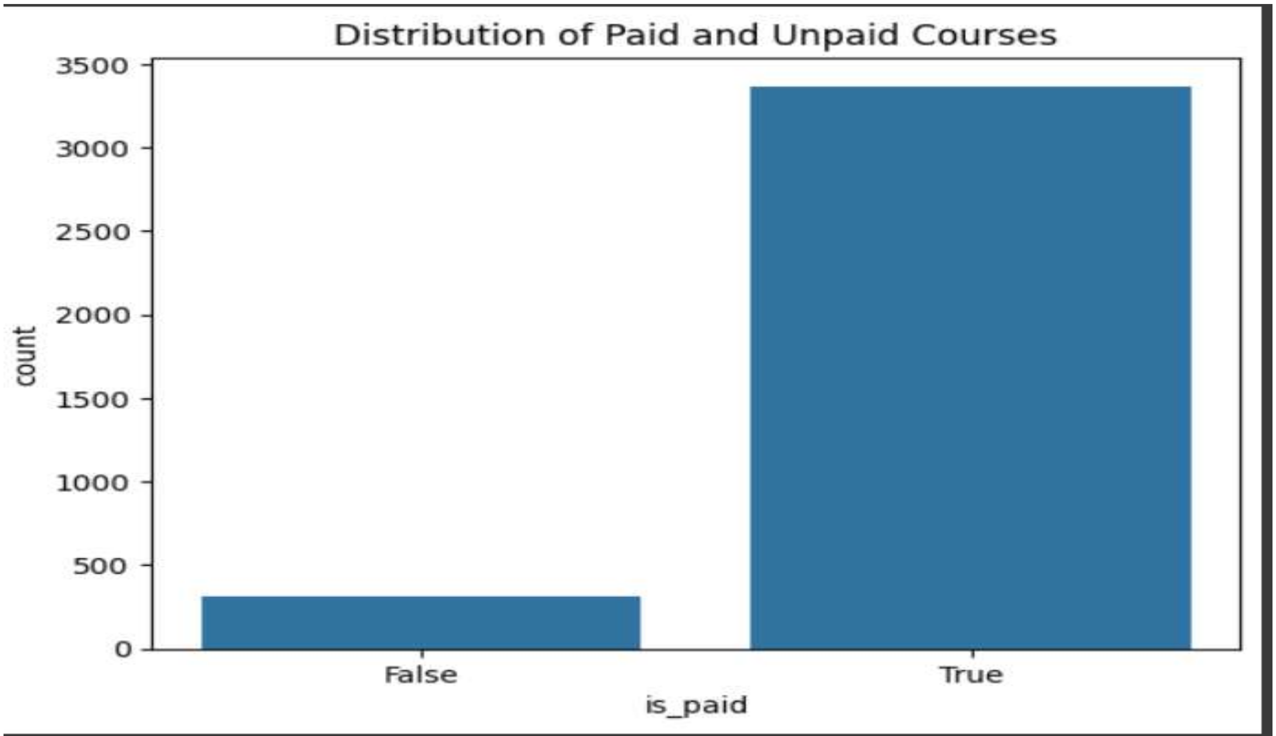
```
--- Collaborative Filtering Recommendations ---
Recommended items for User U12345:
Item C004 with predicted preference score 0.9451
Item C102 with predicted preference score 0.9320
```

Appendix D: Data Visualizations

Figure 3.1 Data Visualizations






Distribution of Paid and Unpaid Courses
3500
3000
2500
2000
1500
1000
500
0
count
False
True
is_paid

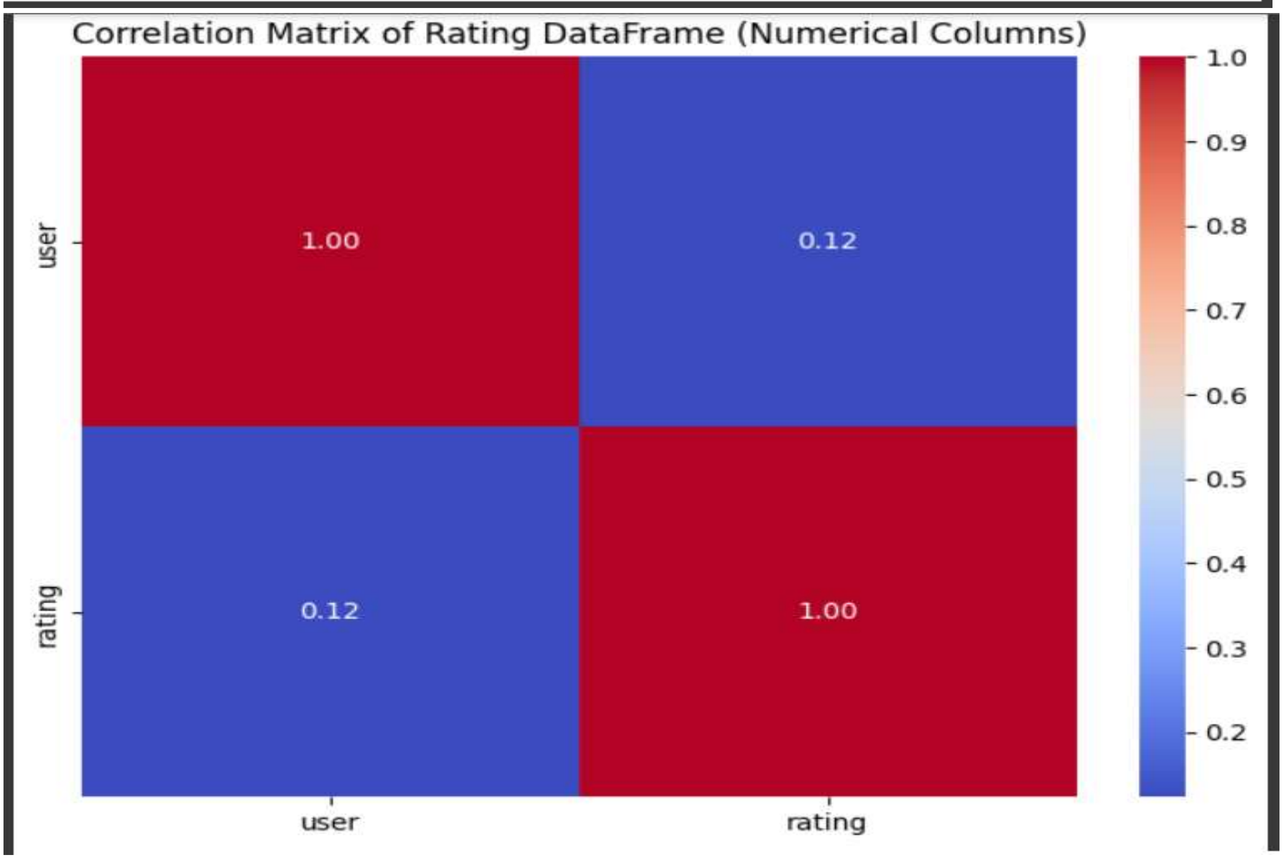

Correlation Matrix of Rating DataFrame (Numerical Columns)
user
rating
1.00
0.12
0.12
1.00
user
rating
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2

Table 3.1 Summary of Existing Studies

Study/Method	Data	Strengths	Weaknesses
Benchmark datasets (OULAD)	Anonymized learner demographics, VLE logs, assessments	Rich interactions; reproducible	UK context; no ratings; limited multilingual support
MOOC surveys / CF + ML/DL	Aggregated across Coursera, edX, XuetangX	Taxonomies of techniques; hybrid trends	Limited coverage of low-resource settings; few explainability studies
Hybrid CF + Clustering	Interaction logs + course attributes	Handles cold-start; skill-course alignment	Requires reliable learner profiling; feature engineering overhead
Context-aware recommenders	Historical interactions + contextual features	Adapts to dynamic learner states	Noisy context; privacy concerns; tuning complexity
Deep/Reinforcement Learning	Sequential MOOC logs	Optimizes completion/mastery	Data-hungry; opaque; difficult in sparse environments
Graph-based / GNN	Interaction graphs + course ontologies	Cold-start support; captures relations	High computational cost; requires domain knowledge
Learning-style infused recommenders	Clickstream + quiz logs	Personalizes presentation	Style inference validity debated; limited generalizability
Ethiopian/local studies	Surveys/interviews with HEIs	Highlights infrastructure & readiness	Few deployed technical recommender systems; limited local datasets