C4U Personalized Course Recommender System

Architectural Decisions Guidelines Executive Summary Report

ISSUE / PROBLEM

We aim that our recommender system can quickly find new interested course, better paving learning path and more learners interacting with more courses. So far, we only have 34,000 users which only 30% have participated more than 10 courses.

Data Source

We choose in the first phase that **batch processing** - data is collected over a period (e.g., daily or weekly) and then ingested into the system before installing real time.

- **OLTP systems** handle real-time transactional processing,

The transactional process in real time

Enterprise Data

To accommodate the current and growth, we need a secure gateway that can handle high traffic, provide secure access to our application, and potentially offer features for API management.

Streaming Analytics

Data types: User-Item Interactions: Implicit feedback (e.g., clicks, views) and explicit feedback (ratings, reviews), Contextual Data: User context (location, device, time) and item context (genre, category) and Item Features: Descriptive attributes of items (e.g., product descriptions, tags).

Data Integration

Select batch processing technologies that offer connectors, APIs, or integration capabilities for seamless interaction with the identified source systems.

Apache Spark with Spark Mllib. Employ collaborative filtering (ALS) or content-based filtering algorithms. Real-time or batch processing to provide recommendations.

Data Repository

Storage for data and model artifacts. Assume 1 TB of storage and Assume 1 TB of data transfer per month.
First 1 GB/month is free.

Discovery and exploration

Our primary metric is in the sales which is increase number of **enrollments and users**. We will used the current dashboard.

Actionable Insights

Apache Spark on Amazon EMR (Elastic MapReduce):

Cluster Configuration:
Master Node: Manages the Spark cluster.
Worker Nodes: Perform data processing and model training.

Application and Data Products

Batch: Recommendations can be **generated periodically** (e.g., daily or weekly). First implement **standardized** then high customization for **public users**.

Security, information governance and systems management

Role-Based Access Control (RBAC): Assign users to predefined roles with specific permissions and privileges.

INSIGHTS/NEXT STEPS

- Total Request per Day = 102,000
- Peak load: 10% during peak hours = 10,200 request per hour
- If User increases to 100,000 daily active users consider using HTAP and Apache Kafka
- Implementation should start in the end of the fourth month of the project