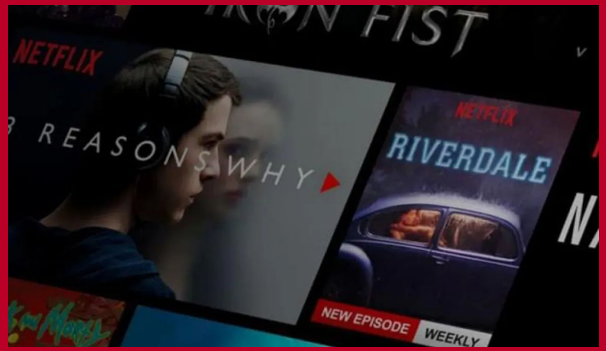
Carnegie Mellon University

11695: ML in Production / AI Eng.
Final Presentation



Team: 09 Overfitting Ocean's Eleven

Presented by

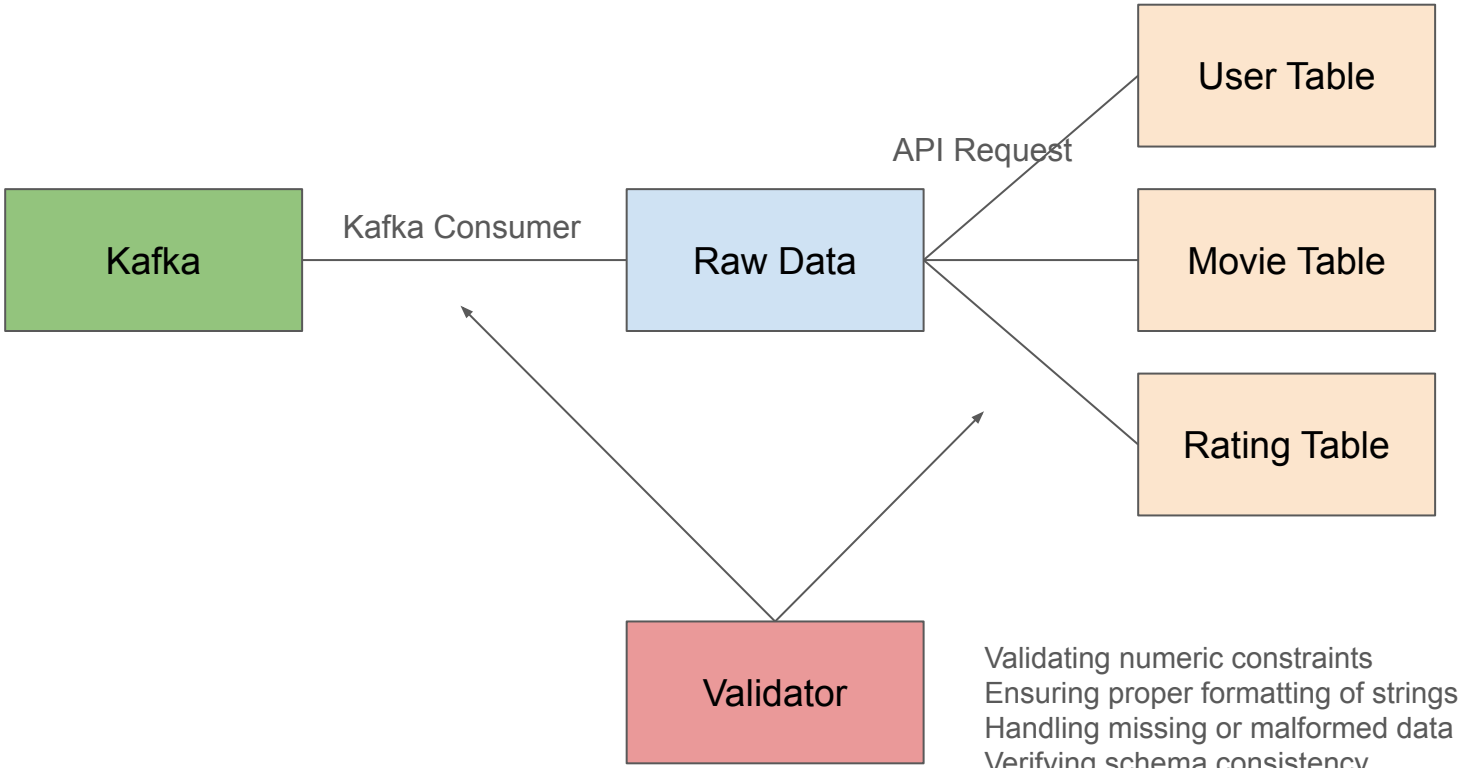
Chujun Tao, Tianyi Cheng, Arjun Choudhry, Zijin Cui, Hin Kit Eric Wong

Reflection on Recommendation Services

Key Components of our Presentation:

- Data Pre-processing
- Model Choice
- Model Update
- Retraining Pipeline Eva
- Containerization
- Provenance
- Teamwork

Data Preprocessing

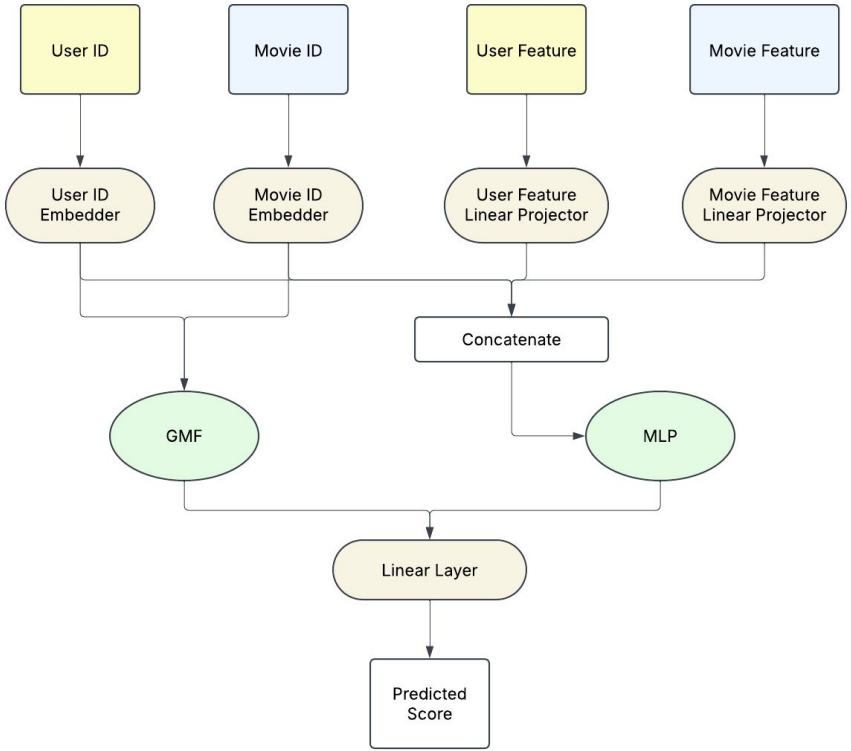


Incremental store for regular retrain

Validating numeric constraints
Ensuring proper formatting of strings
Handling missing or malformed data
Verifying schema consistency

Carnegie Mellon University

Model Architecture - Feature Neural Collaborative Filtering



User Features:
- Age
- Gender
- Occupation

Movie Features:
- Popularity
- Runtime
- Vote Count
- Vote Average
- Adult
- Original Language
- Genres

Carnegie Mellon University

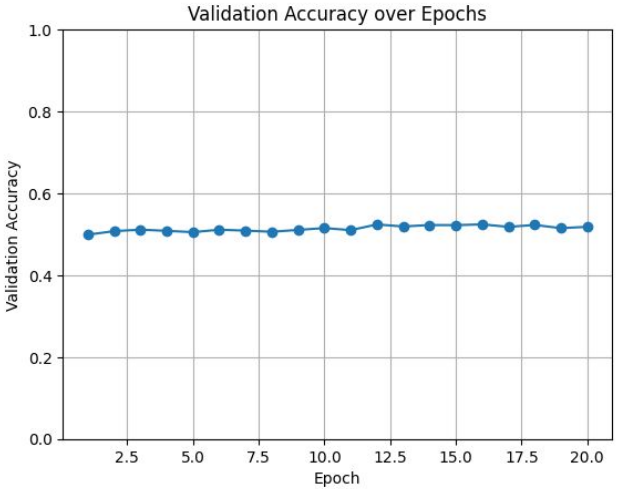
Model Design

Inference

- 1 User v.s. all Movies
  - pairwise rating prediction
  - batch process -> high efficiency
- Unknown User (cold start)
  - num. users fixed at training
  - for data of N users, train on N+1 num. users
  - User ID is N+1 for all unknown users

Hyper-Parameter

Train 1 epoch -> efficiency



Carnegie Mellon University

Model Update

Automated Data Collection

- Collect Kafka ratings
- Append new users to all user info
- Append new movies to all movie info

Automated Model Training

- After data collection
- Use updated data from fixed directory
- Re-train new model on new data
- Save trained model and related info

Central Model Storage (MinIO)

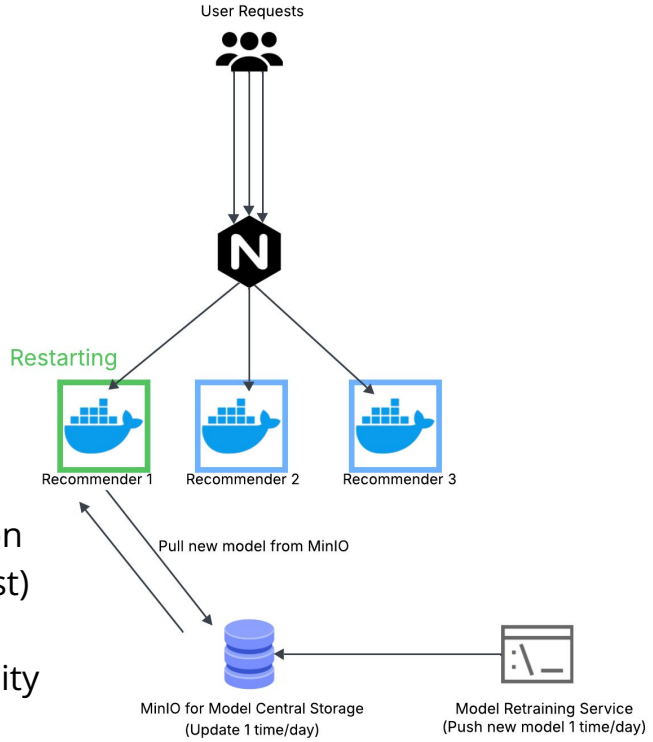
- Stores versioned models
- Provides unified access point for inference services
- Ensures model consistency and traceability

Model Update and Deployment

- Inference service pulls the latest model from MinIO on restart
- Reloads model into memory at startup
    - Advantage: No need to rebuild container
- Restart can be manual or triggered by Cronjob automatically once per day.

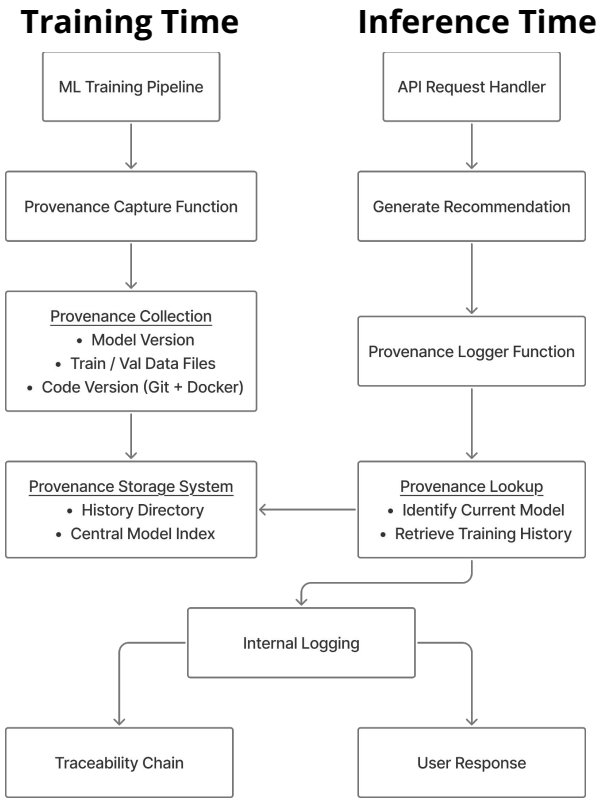
Containerization and Continuous Integration (CI)

- **Tools:** Docker
- **Containerization Solution:**
  - Flask recommender service containers
  - Nginx reverse proxy
  - Orchestration: Docker Compose
- **CI for Containerization**
  - Github Actions
  - Two workflows:
    - Build Docker Images
      - Verify container image creation
    - Unit Test and Integration Test (Pytest)
      - Verify code quality and functionality to ensure reliability



Carnegie Mellon University

Provenance



Training Time

- Captures model version, training data, and code / pipeline version
- Stores detailed history in structured JSON files and central index

Inference Time

- Generates recommendations using active models
- Retrieves and logs complete provenance metadata internally
- Returns only recommendations to users

Key Benefits

- Full internal traceability chain for debugging and auditing
- Clean separation between internal tracking and user experience
- Answers critical questions: which model, which code, which data?

Carnegie Mellon University

Reflection on Teamwork



Carnegie Mellon University

Thank You!