Federated & Feature-Fused Movie Recommendation System (FF-MovieLens)

Leveraging Federated Learning, Matrix Factorization, and Neural Models for Privacy-Preserving Personalization

Navya Manjari Uppaluri
Faculty of Computer Science, Dalhousie University
nv574116@dal.ca

DALHOUSIE UNIVERSITY

The Movie Recommendation Paradox

- **Problem:** Too much content, not enough clarity
- **Challenge:** Platforms collect two very different types of feedback:
  - **Explicit:** Ratings, reviews — accurate but sparse
  - **Implicit:** Clicks, watch time — abundant but noisy
- **Result:** Users face endless scrolling without confident decisions

Motivation

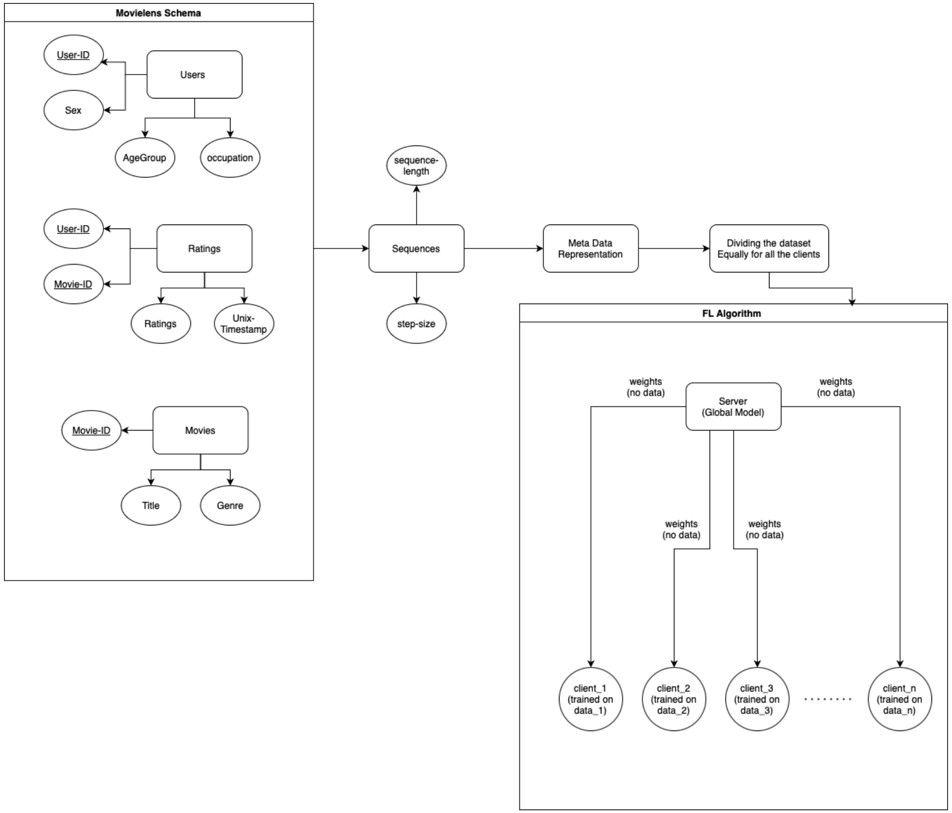
- **Data Privacy Concern:** Centralized recommenders collect and store sensitive user data
- **Federated Learning (FL):** Train models without centralizing raw data
- **Goal:** Build a **privacy-preserving movie recommender** with minimal accuracy loss
- **Approach:** Combine FL with **neural networks** to learn rich user–item patterns
- **Benchmark:** Evaluate on **MovieLens-1M**, a widely used dataset in recommender research

The Core Challenge

- Combine **rich but sparse** and **abundant but noisy** data
- Preserve **user privacy** while improving personalization
- Handle **cold-start** and **data heterogeneity** in a real-world scenario

Our Solution: Federated + Feature-Fused Architecture

- **Federated Learning Layer:**
- Keeps raw user data local
- Aggregates model weights (FedAvg Adam)
- **Feature Fusion Layer:**
- Uses user demographics (age, gender, occupation)
- Uses movie metadata (genres)
- **Model Components:**
- Baselines (Popularity, User-Avg)
- Matrix Factorization (SVD)
- Neural Embedding Models



Dataset Overview (MovieLens-1M)

•6,040 users, 3,883 movies, 1M+ ratings

•Demographics: Age, gender, occupation

•18 genres, ratings from 1939 onwards

•Dataset sparsity: 95.8%

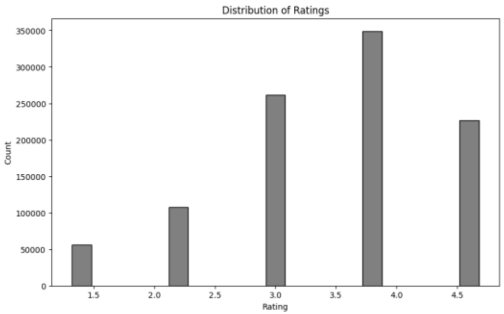
•**Pre-processing:**

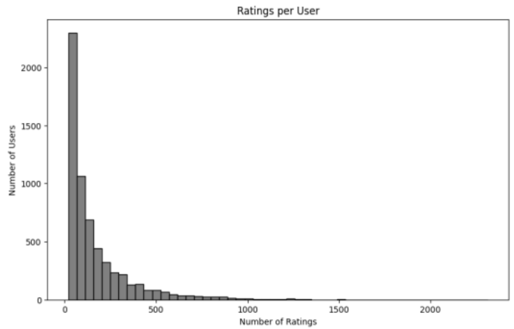
- Sequencing movie IDs & ratings
- Encoding genres, occupation, gender, age
- Creating non-IID partitions for clients

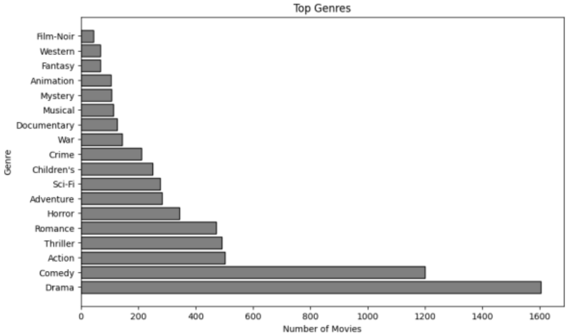
Name	Statistics
Number of users	6040
Number of movies	4052
Number of reviews	1000209
Click/browse	6976551
Review	354016
Favorites	72604
Like	244740
Total user actions	7647911
Dataset sparsity	0.958

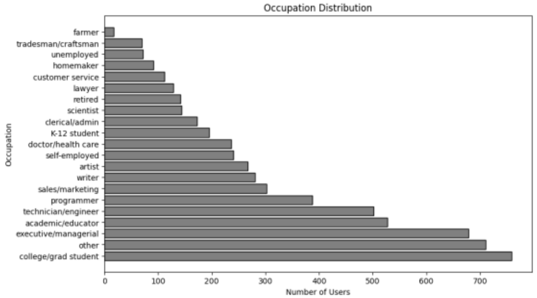
EDA Highlights

- Ratings distribution: Skewed towards higher ratings (bias towards liked movies)
- Genre popularity: Top genres are Drama, Comedy, Action
- Sparsity: ~95% of user–item matrix is empty
- Active users dominate dataset, tail users have few ratings









Baseline Models

•**Popularity-Based Prediction:** Recommend highest-rated items globally

•**User-Average Prediction:** Predict user's average rating for unseen items

•**Cold-Start Evaluation:** Test on users/items with no prior interactions

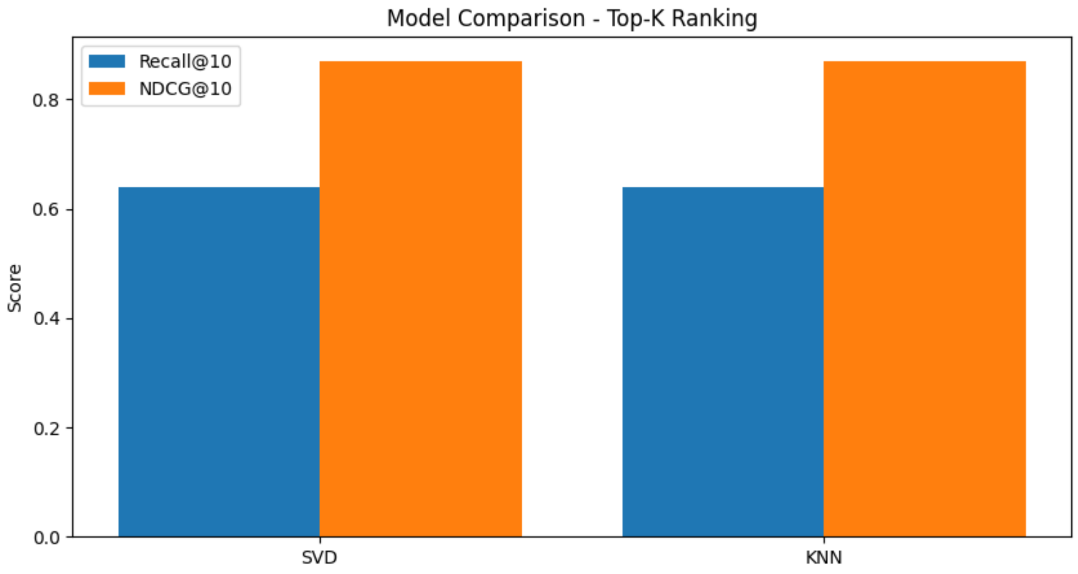
•**Performance:**
– Popularity MAE: 0.7788
– User Avg MAE: 0.8234

Feature Ablation Test

- Purpose: Measure impact of removing features
- Observations:
  - Removing popularity feature → MAE ↑
  - Removing user-avg feature → MAE ↑
  - No features → MAE very high (~3.58)
- Conclusion: Each feature contributes significantly

Advanced Models

- **SVD Matrix Factorization**
  - Learns latent user/item factors
  - MAE: 0.6850
- **KNN Collaborative Filtering**
  - User-based MAE: 0.7713
  - Item-based MAE: 0.7794
- **Top-K Ranking Metrics**
  - Recall@10: 0.6398
  - NDCG@10: 0.8709



Neural Model Architecture

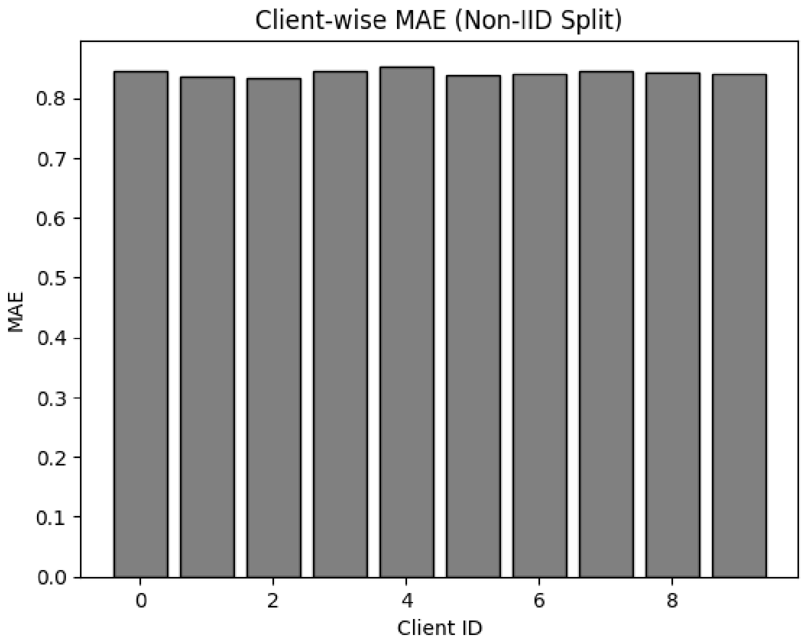
- **Inputs:**
  - **User ID and Movie ID embeddings**
  - **Encoded genres, occupations, age, and gender**
- **Hidden Layers:**
  - **256 → 128 → 1 units**
- **Activation:** ReLU
- **Regularization:** Dropout
- **Output:** Predicted rating

Model Architecture – FFNN

- **Framework:** PyTorch FeedForward Neural Network (FFNN)
- **Embedding Layers:**
  - **User IDs**
  - **Each of 4 movie IDs in the sequence**
- **Forward Pass:**
  - **Concatenate embeddings**
  - **Flatten → Linear → ReLU → Dropout → Linear**
- **Output:** Probability distribution across all movie IDs
- **Loss Function:** CrossEntropyLoss
- **Optimizer:** Adam
- **Training:** 10 epochs

Federated Learning Setup & Aggregation Logic

- **Dataset Split:** User IDs divided into 10 clients
- **Local Training:**
  - Each client trains a local copy of the FFNN
  - Fixed number of local epochs
- **Aggregation Logic (FedAvg):**
  - Server collects model weights from all clients
  - Averages parameters (weights)
  - Updates global model
- **Advantages:**
  - User data remains local
  - Improved personalization
  - Reduced risk of data breaches



Evaluation Metrics

•**Metrics Used:**

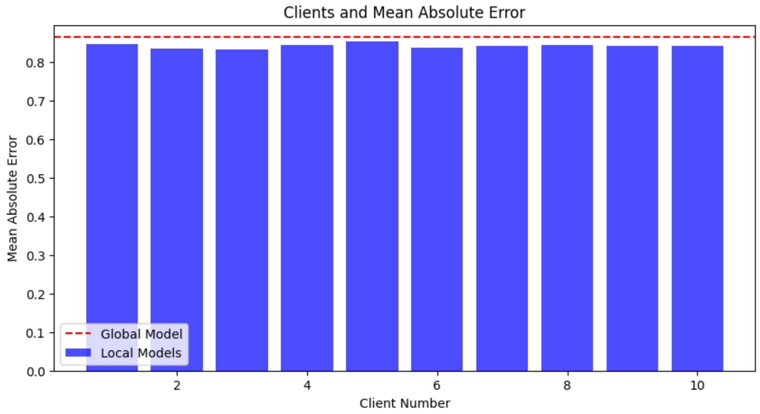
- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- R^2 Score
- Mean Absolute Percentage Error (MAPE)

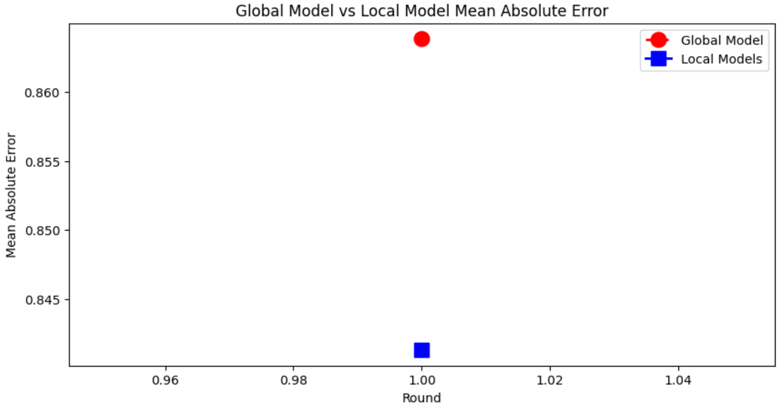
•**Evaluation conducted:**

- On individual client models
- On final global model (after aggregation)

Experimental Results

- Federated Feed-Forward NN (10 clients):
  - Best MAE: 0.8378
  - R^2 ~ 0.13
- SVD outperforms KNN in MAE
- Top-K ranking:
  - SVD Recall@10 ~ 0.64
  - NDCG@10 ~ 0.87





Results & Comparisons

Client Wise Performance table

Client	MAE	MAPE	R^2	MSE	RMSE
1	0.861271	34.193677	0.121443	1.110641	1.053869
2	0.844557	33.979844	0.128833	1.083926	1.041118
3	0.838203	34.173737	0.130822	1.078016	1.038275
4	0.843004	34.873277	0.123515	1.094560	1.046212
5	0.837860	35.227289	0.118945	1.099636	1.048635
6	0.856534	33.510565	0.120854	1.095998	1.046899
7	0.850350	33.967662	0.122866	1.093216	1.045570
8	0.844025	34.068765	0.130675	1.085979	1.042103
9	0.858443	33.897860	0.112239	1.105887	1.051612
10	0.845914	34.077291	0.127015	1.088638	1.043378

Personalized Case Study

- Example User: ID 1841
- Top 5 Recommendations:
  - Babe (1995)
  - Pulp Fiction (1994)
  - Shawshank Redemption (1994)
- Matches user’s highly rated genres

Key Technical Contributions

- **Privacy-Preserving FL**
  - No raw data sharing
- **Feature Fusion**
  - Metadata + Ratings
- **Robust Evaluation**
  - MAE, Recall@10, NDCG@10, Ablation

Challenges Faced

- **Data Imbalance:** Some clients have fewer samples
- **Synchronization:** Ensuring consistent training configurations across clients
- **Resource Constraints:** No deployment or parallelism used

Conclusion

- Successfully combined **FL, MF, and deep models**
- Improved personalization while **preserving privacy**
- Strong results on MovieLens-1M
- Applicable to other domains beyond movies

Limitations & Future Work

•Limitations

- Cold start problem for new users or items
- Assumes static user preferences over time
- Limited interpretability of deep learning models

•Future Enhancements

- Integrate temporal modeling (RNN / Transformer) for evolving preferences
- Apply multi-modal fusion (e.g., text, images) for richer recommendations
- Develop explainable AI methods for transparent decision-making

References

- Harper, F. M., & Konstan, J. A. (2016). *MovieLens 1M Dataset.* GroupLens Research Project.
- McMahan, B., et al. (2017). *Communication-efficient learning of deep networks from decentralized data.* In *AISTATS.*
- Ammad-Ud-Din, M., et al. (2019). *Federated collaborative filtering for privacy-preserving personalized recommendation systems.* arXiv:1901.09888.
- Li, L., et al. (2020). *A review of applications in federated learning. Computers & Industrial Engineering*, 149, 106795.
- Kairouz, P., et al. (2021). *Advances and open problems in federated learning. Foundations and Trends in Machine Learning*, 14(1–2), 1–210.


THANK YOU