Group 19 Topic: MovieLens Recommender System

Collaborative Filtering and SVD in KDD Pipeline

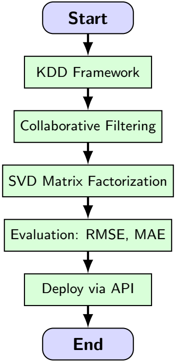
Tran Viet Cuong, Doan Duc Hoang, Nguyen Vu Gia Huy,
Duong Dam Lam, Trinh Quang Minh, Nguyen Quoc Viet

University of Science and Technology of Hanoi
Vietnamese France University
ICT Lab

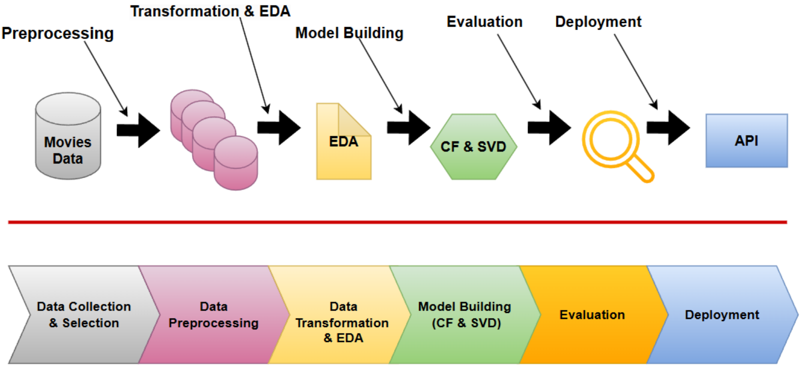
March, 2025

Objectives & Introduction

- Build a scalable movie recommender using Collaborative Filtering + SVD
- Follow KDD process: from data preprocessing to deployment
- Evaluate using RMSE and MAE
- Tackle challenges: data sparsity, cold-start problem



System Architecture



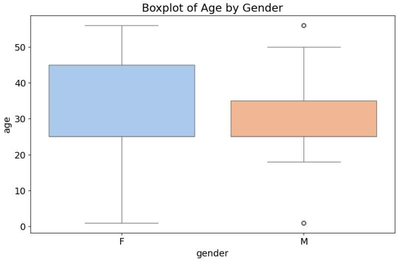
KDD Pipeline integrating MovieLens, CF, SVD, and API

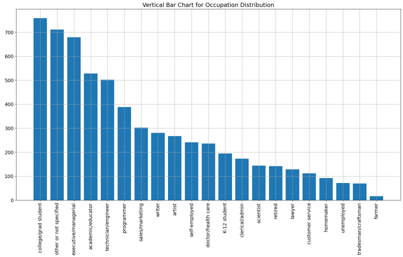
Data Preprocessing

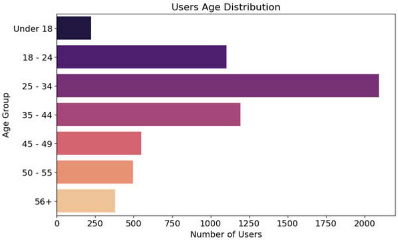
The MovieLens 1M dataset consists of three main files:

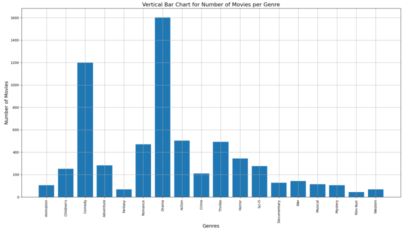
- **users.dat**
  - **Records:** About 6,040 users.
  - **Features:** UserID, Gender, Age, Occupation, and Zip-code.
  - **Purpose:** Provides demographic information.
- **movies.dat**
  - **Records:** Approximately 3,883 movies.
  - **Features:** MovieID, Title, and Genres.
  - **Purpose:** Contains basic movie details for categorization.
- **ratings.dat**
  - **Records:** Nearly 1,000,209 ratings.
  - **Features:** UserID, MovieID, Rating, and Timestamp.
  - **Purpose:** Records user–movie interactions.

Data Visualization









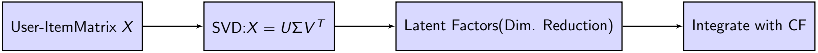
Collaborative Filtering (CF)

User-Item Matrix → Compute Similarities(e.g., Cosine) → Select Top-kSimilar Movies → Predict / Recommend

Explanation:

1. **User-Item Matrix:** Rows are users, columns are movies, and cells represent ratings.
2. **Compute Similarities:** Calculate how closely movies resemble each other based on user ratings.
3. **Select Top-k Similar Movies:** Pick the most similar movies to the target title.
4. **Predict / Recommend:** Generate personalized predictions or directly recommend the Top-N similar titles.

Singular Value Decomposition (SVD)



- **Key Purpose:** Reveal latent patterns in the user-item matrix.
- **Process:**
  - *Center Data* (subtract mean ratings).
  - *Dimensionality Reduction* (keep top-p factors).
- **Outcome:** SVD + CF did not improve RMSE/MAE in our tests.
- **Note:** Direct CF outperformed SVD, likely due to sparse data or insufficient tuning.

Results and Evaluation

Why RMSE and MAE?

- **RMSE (Root Mean Squared Error):** Penalizes larger errors more heavily; highlights big deviations.
- **MAE (Mean Absolute Error):** Reflects the average magnitude of prediction errors; straightforward to interpret.

Method	RMSE	MAE
CF Only	0.9184	0.7344
SVD + CF	2.7405	2.4679

Table: Comparison of CF vs. SVD + CF

Key Observations:

- CF alone achieves lower RMSE and MAE, indicating better accuracy.
- SVD + CF may require more parameter tuning or denser data to outperform CF.

Sample SVD Recommendations:

- **Film ID 45:** Similar titles: 322, 1120, 537, 52, 1885
- **Film ID 60:** Similar titles: 2, 1848, 3489, 1702, 362

System Strengths & Limitations

Strengths:

- CF provides more accurate predictions (low RMSE/MAE).
- SVD helps reduce dimensionality and noise, although CF alone performed better on our dataset.
- Scalable approach for handling large user-item matrices.

Challenges:

- Cold-start problem for new users/items.
- Data sparsity can reduce model effectiveness.
- Parameter tuning (e.g., number of latent factors) can be non-trivial.
- No deep learning methods yet; could explore advanced techniques.
- Deployment: Requires an API or real-time pipeline for practical use.

Conclusion & Future Work

- CF improves movie recommendation accuracy
- CF + SVD show a lack of accuracy in movie recommendations.
- Preprocessing and KDD steps ensure reproducibility
- **Next Steps:**
  - Hybrid recommender with content-based methods
  - Use GNNs, DL for advanced personalization
  - Real-time feedback and retraining
  - Deploy a real-time RESTful API

Thank You!

Questions and Answers?