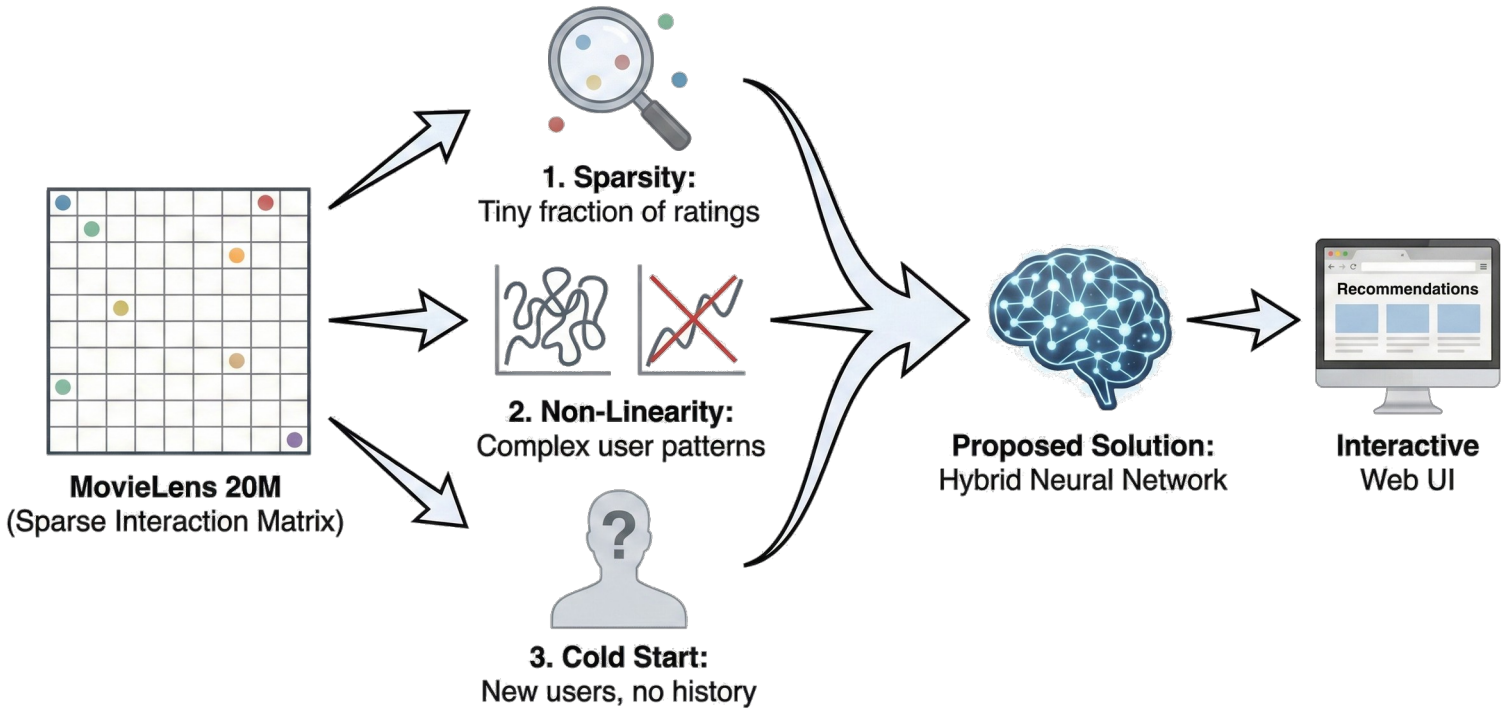
Movie Recommendation

Théodore Billotte, Louis Rollet, Vincent Montero Fontaine
Department of Computer Science and Technology

Introduction & Problem Statement

Objective: Build a robust recommendation engine capable of predicting user ratings and solving the "Cold Start" problem.

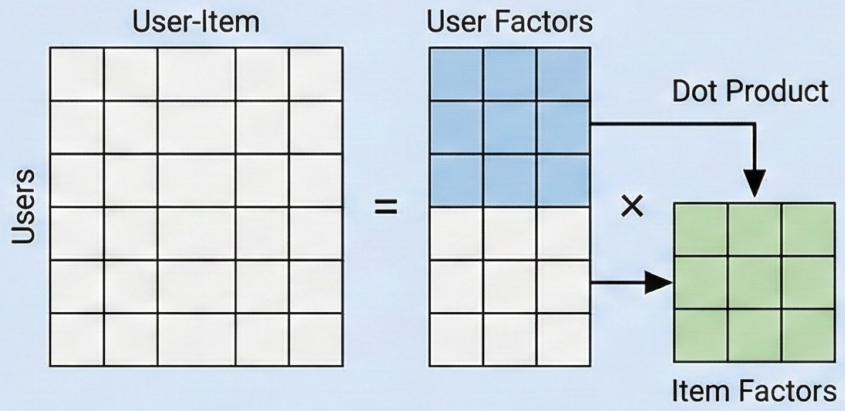


Proposed Solution: A multi-model benchmarking approach culminating in a **Hybrid Neural Network** deployed via an interactive Web UI.

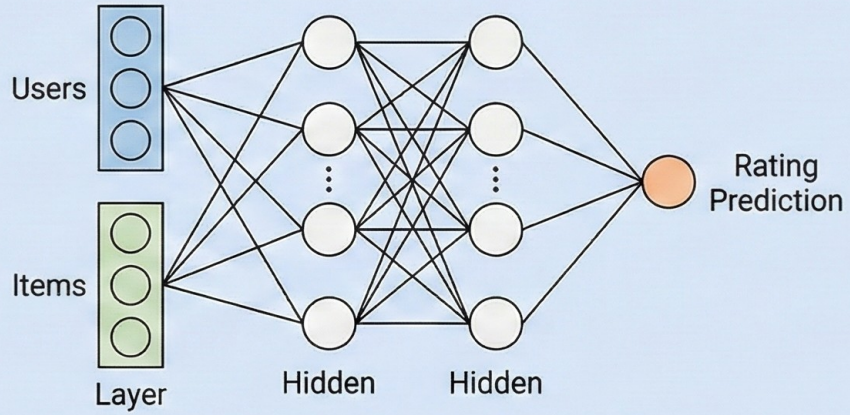
Harper, F. M., & Konstan, J. A. (2015). *The MovieLens Datasets: History and Context.* ACM Transactions on Interactive Intelligent Systems (TiiS).

Methodology Overview

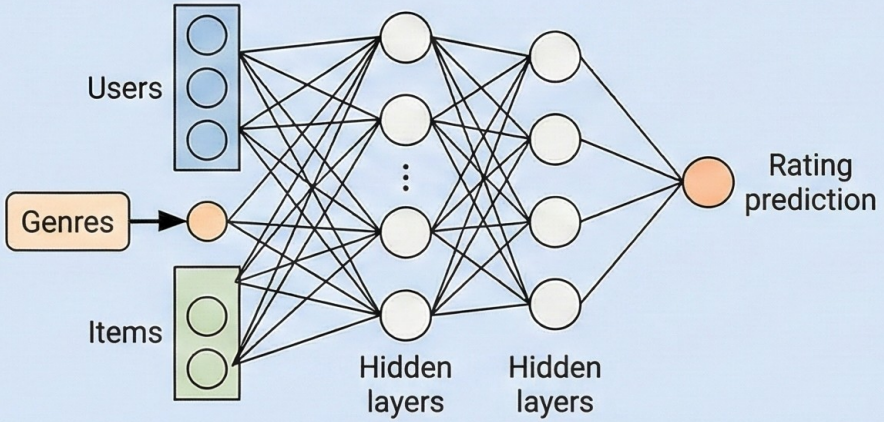
Model 1: Matrix Factorization (Baseline)



Model 2: Neural Collaborative Filtering (NCF)



Model 3: Hybrid NCF (Extension with Genres)



Model 4: LLM Agent



Model 1 - The Baseline (Matrix Factorization)

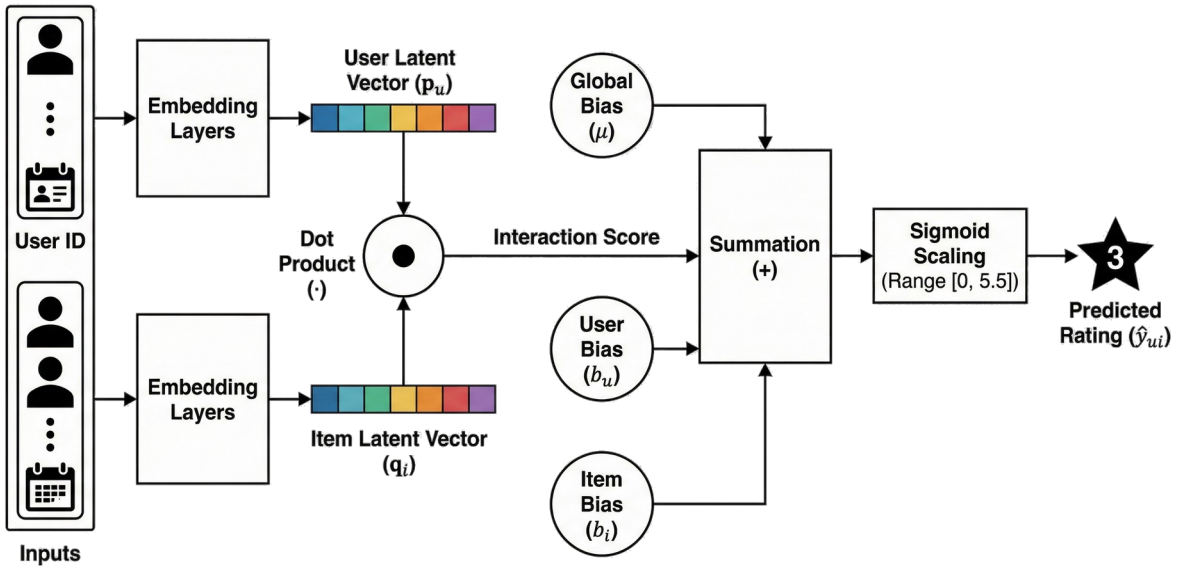
Concept: Decomposes the interaction matrix into latent feature vectors for Users and Items.

Methodology: Standard SVD-style factorization implemented in PyTorch.

$$\hat{y}_{ui} = \mu + b_u + b_i + \mathbf{p}_u^T \mathbf{q}_i$$

Where:

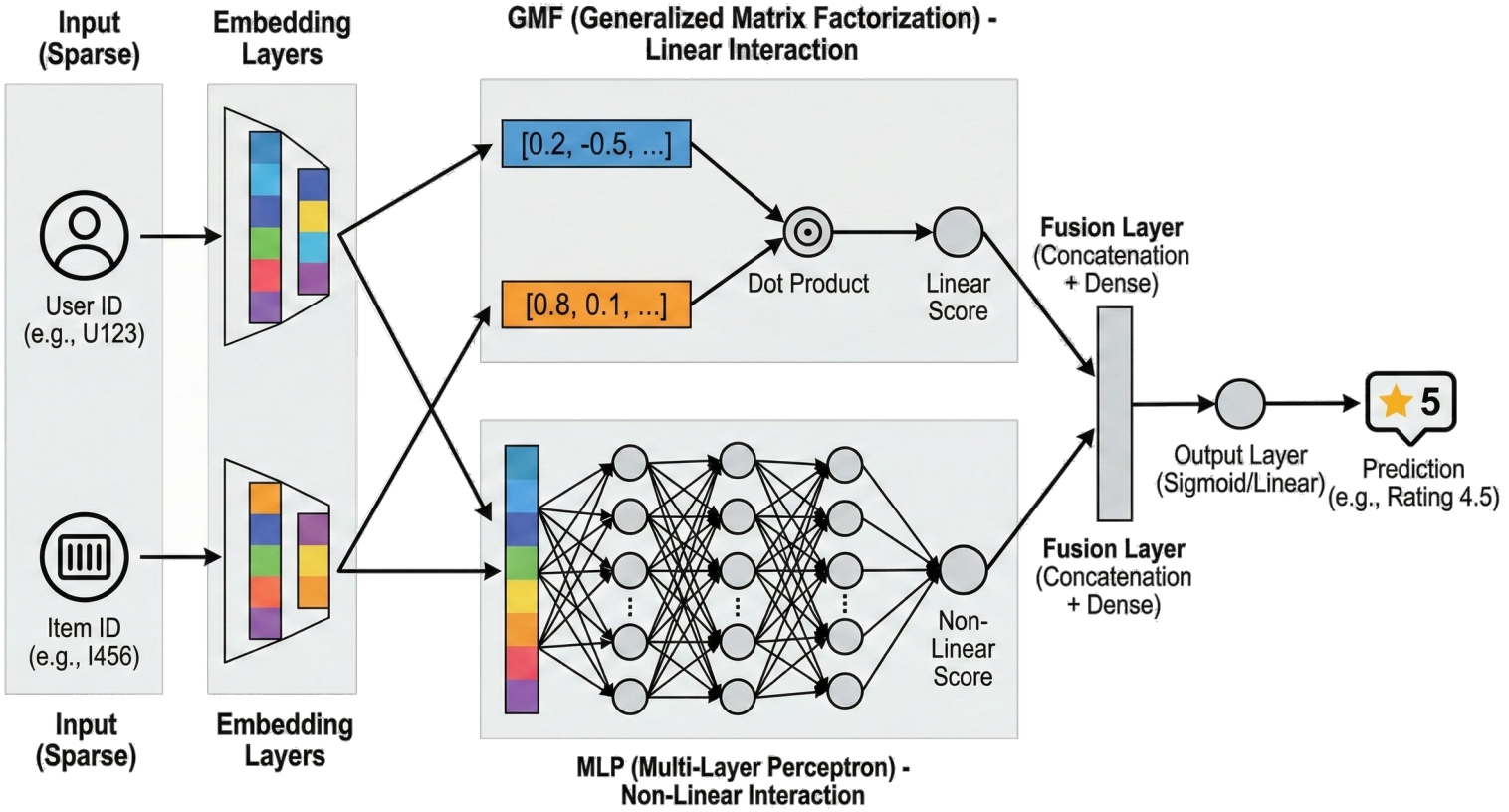
- μ: Global bias
- b_u, b_i: User and Item biases
- $\mathbf{p}_u, \mathbf{q}_i$: Latent vectors for user u and item i



Pros: Fast and interpretable.
Cons: limited linearity (underfits complex patterns).

Koren, Y., et al. (2009). *Matrix Factorization Techniques for Recommender Systems.* IEEE Computer.

Model 2 - Neural Collaborative Filtering (NCF)



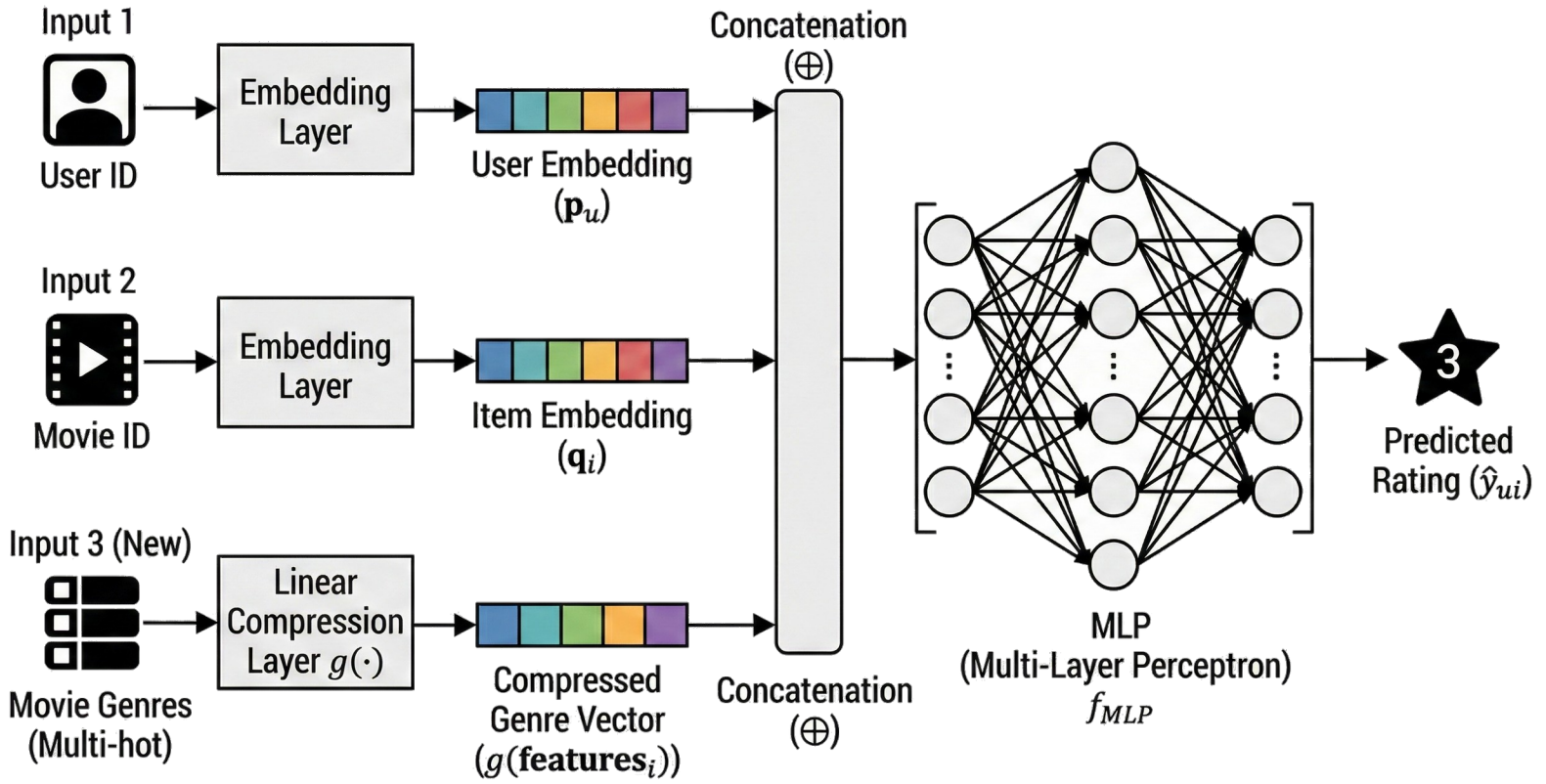
Motivation: Addressing the limitations of the dot product (linear combination) in Matrix Factorization.

Advantage: Significantly lower RMSE than baseline (~0.82 vs ~1.16) due to higher model capacity.

He, X., et al. (2017). *Neural Collaborative Filtering.* Proceedings of the 26th International Conference on World Wide Web (WWW).

Model 3 - Proposed Hybrid NCF

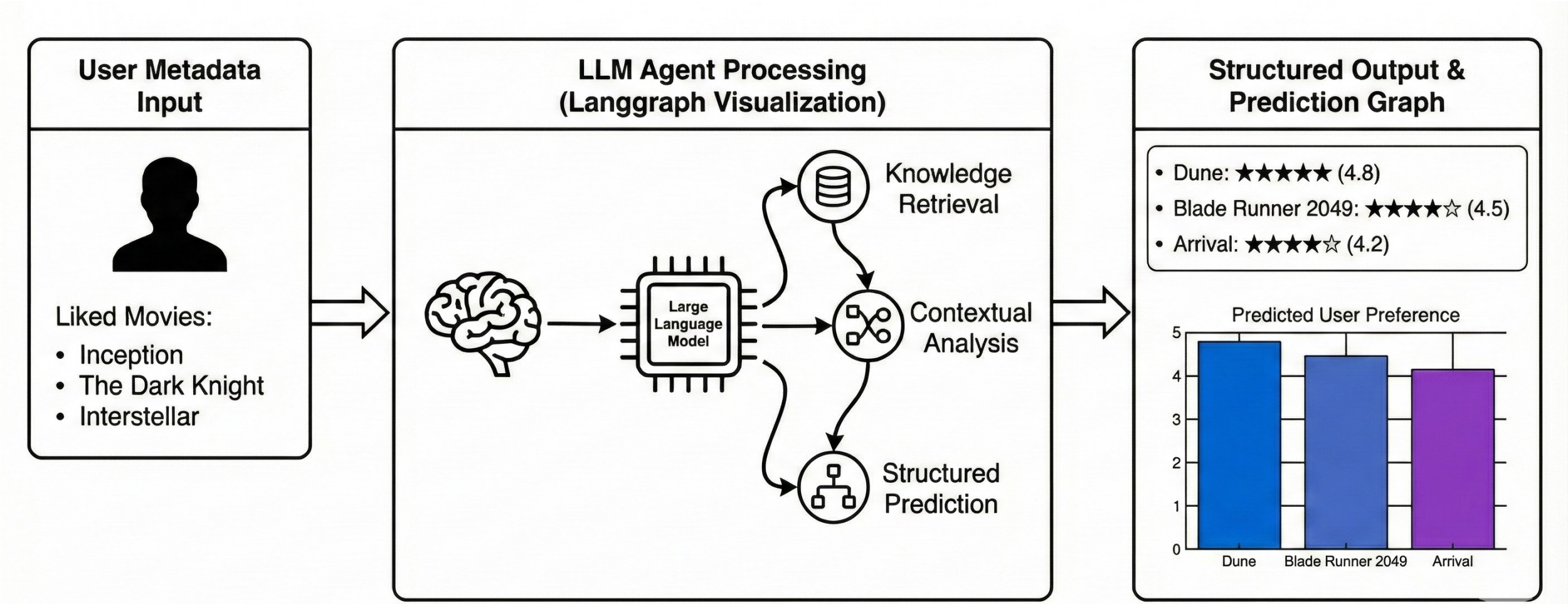
Innovation: Extending NCF to include Content-Based Features (Genres).



- Allows the model to "understand" item nature, not just ID.
- Essential for interpreting why a recommendation was made (e.g., "User likes Sci-Fi").

Burke, R. (2002). Hybrid Recommender Systems.Secondary
Wang, H., et al. (2015). Collaborative Deep Learning for Recommender Systems.

Model 4 – LLM Agent



Liu, J., et al. (2023). LLM-Rec: Personalized Recommendation via Prompting Large Language Models. arXiv preprint arXiv:2307.15780v3.

Experimental Setup & Training

Framework & Hyperparameters:

- Framework: PyTorch
- Optimizer: Adam (Adaptive Moment Estimation)
- Loss Function: MSE (Mean Squared Error)
- Batch Size: 1024 (Efficient GPU utilization)
- Epochs: 5 (Early convergence observed)

PyTorch

Mean Error Squared

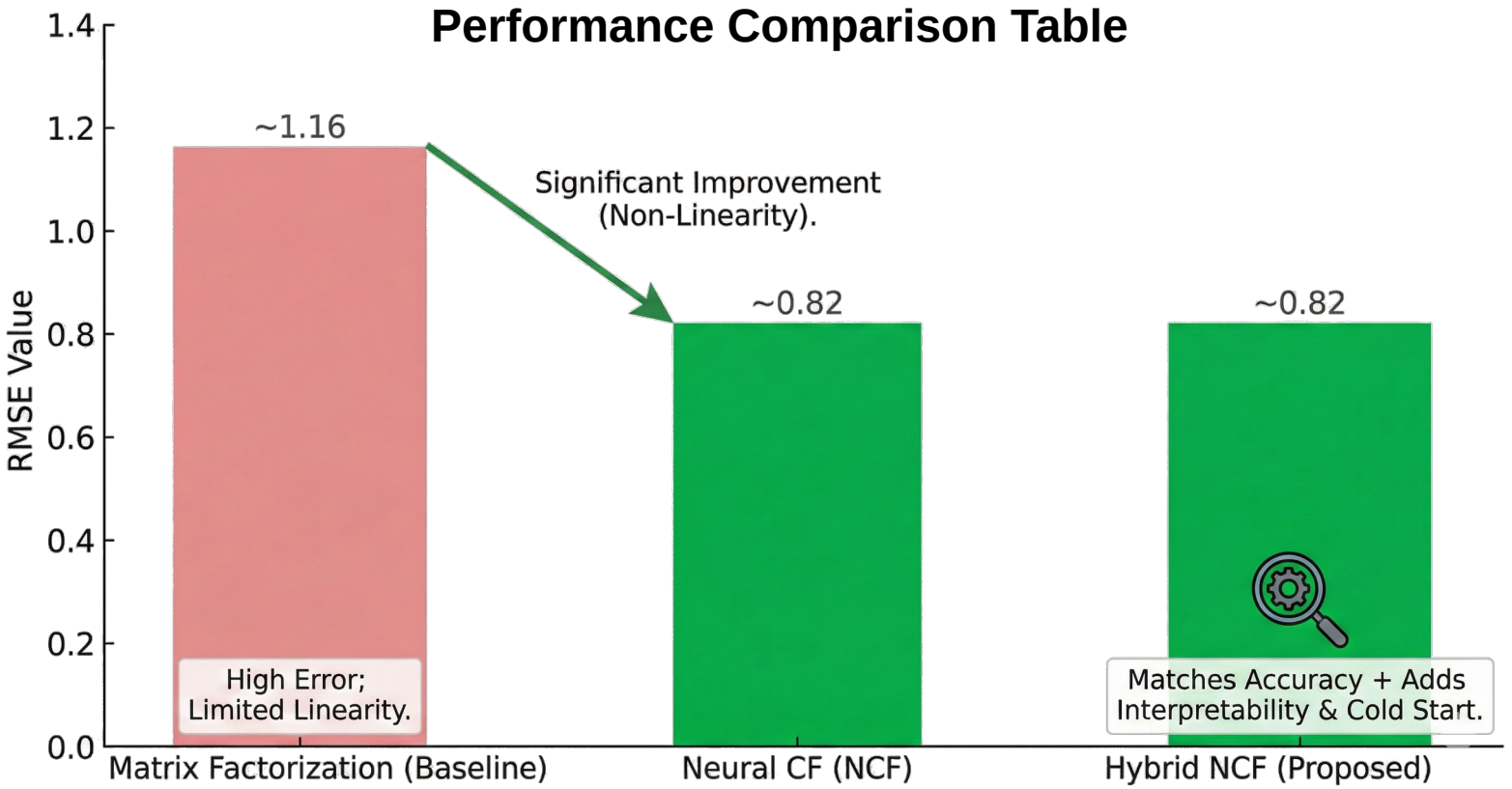
$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Data & Reproducibility

- Data Splitting: Train/Test split applied to interactions
- Reproducibility: Weights saved as .pth checkpoints in checkpoints/ directory

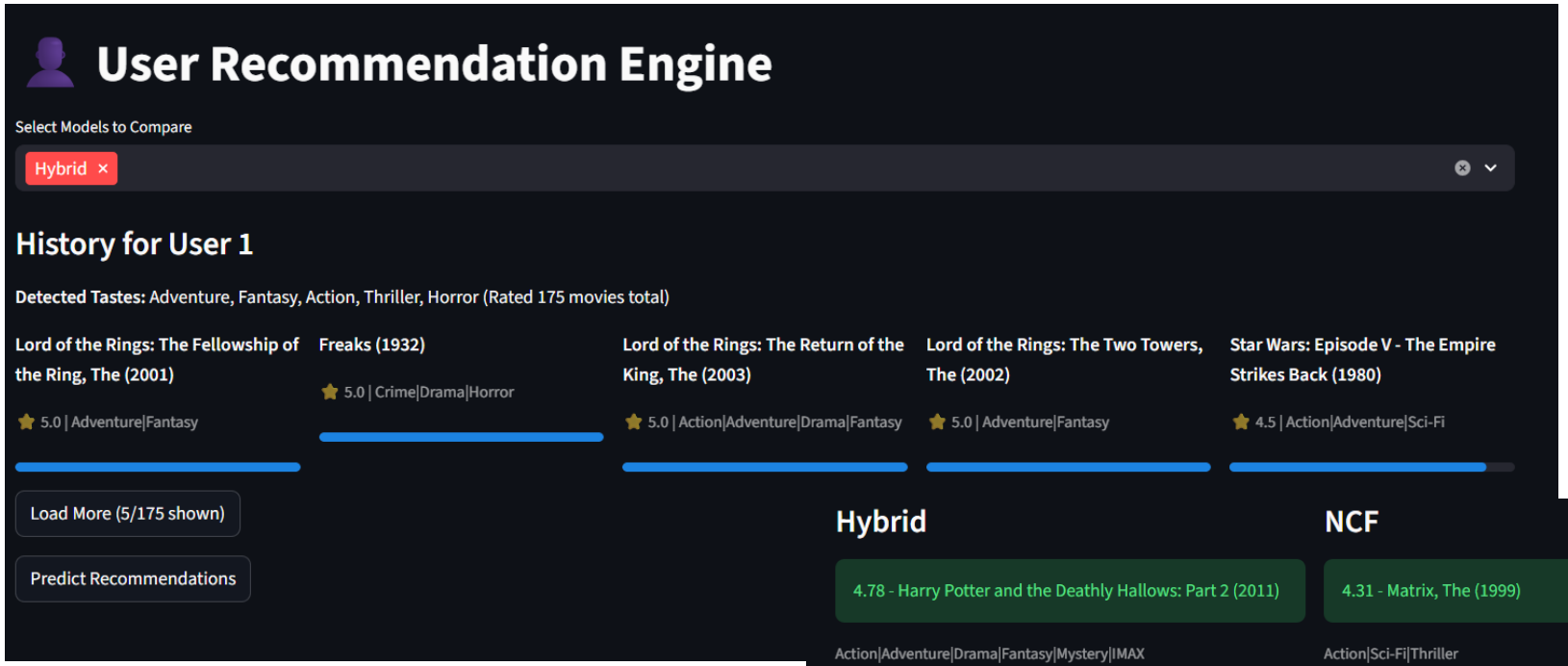
Gunawardana, A., & Shani, G. (2009). A Survey of Accuracy Evaluation Metrics of Recommendation Tasks.

Results & Evaluation



- Neural models drastically outperform the linear baseline.
- Adding genres (Hybrid) did not lower RMSE further (due to latent factors already capturing genre info implicitly) but enabled Cold Start features.

Application - Interactive Dashboard (Streamlit)



Choose a User

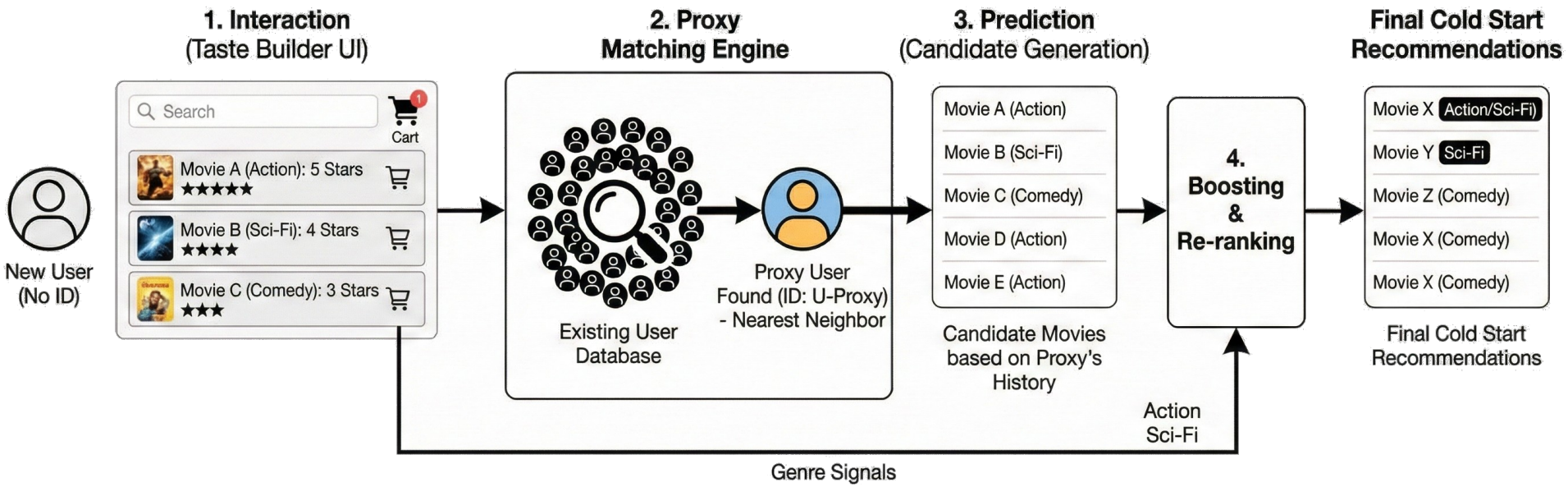
Predicted liked movies using all models



Solving the "Cold Start" Problem

The Challenge: How to recommend movies to a new user with no ID or history?

Our Taste Builder approach



Conclusion & Future Work

Summary:

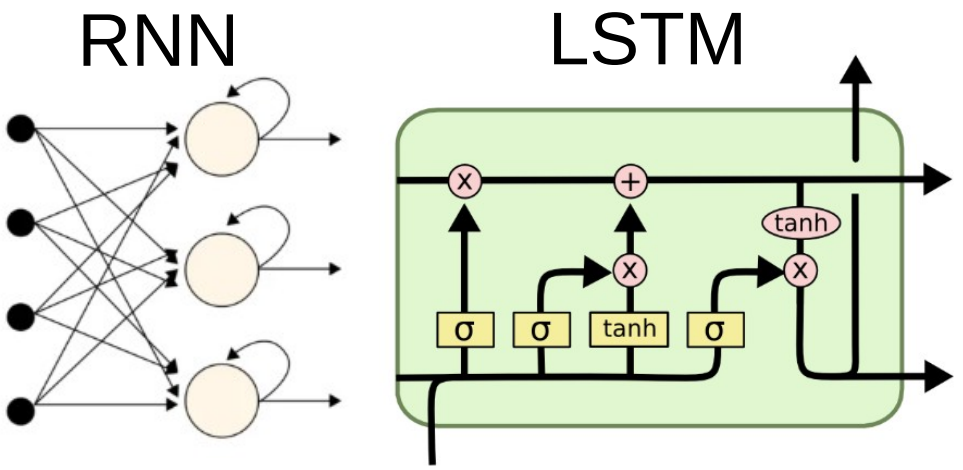
- Successfully benchmarked three architectures.
- Demonstrated the superiority of Neural approaches over Matrix Factorization.
- Built a production-ready UI bridging mathematical embeddings with user needs.

Limitations:

- Dataset size (20M) requires significant RAM.
- Hybrid model complexity increases inference time slightly.

Future Improvements

- Integrate Temporal Dynamics (Time-based RNNs/LSTMs).
- Use Auto-Encoders for better feature compression.
- Deploy model via Docker/Cloud.



Thank you!

Théodore Billotte, Louis Rollet, Vincent Montero-Fontaine
Department of Computer Science and Technology