

# Netflix Movie Recommender Systems

*Members: Steven Nguyen, Mohammed Abdul, Cole Schaffer*

## 1. Introduction, Dataset, and Preprocessing

Our project tackles the challenge of “information overload” on Netflix, where users are confronted with hundreds of thousands of movies, leading to indecision and missed opportunities for discovery. The goal was to design a personalized recommendation system that predicts what each user will enjoy based on their historical ratings, making the library accessible.

We used the Netflix Prize dataset, a gold standard for recommender research, containing over 100 million ratings from ~480,000 users on 17,700 movies, with scores from 1 to 5. One core challenge is sparsity: about 98% of the matrix is empty, with most users rating only a fraction. Ratings also follow a long-tail distribution, with a few popular movies receiving tens of thousands of ratings, while many receive very few.

During exploration, we observed that most active users ( $\geq 5$  ratings) are the majority, and only 1.5% are “cold-start” users. On the item side, 8.6% suffer from limited feedback, making cold-start more severe for items. To ensure fair evaluation, we split data temporally: earliest 80% for training, next 10% for validation, latest 10% for testing. We also enforced that all items in validation or test appeared in training, preventing leakage and supporting robust model comparison. These steps ensured models were trained realistically despite sparsity and the long-tail distribution.

## 2. Baseline Models and Collaborative Filtering Approaches

We started with building baseline models as simple benchmarks to establish initial performance. The Global Mean predicts ratings as the dataset average, ignoring user or item differences. The Movie Mean predicts each movie’s average rating, capturing popularity bias, while the User Mean reflects each user’s average behavior. Both reduce RMSE and MAE compared to the global mean, but errors remain high, showing these naive methods are too simplistic for strong recommendations, highlighting the need for advanced methods.

Item-Based Collaborative Filtering (Item-CF) builds on this by utilizing relationships between movies. The model assumes that if a user enjoyed certain movies, they are more likely to enjoy others with similar rating patterns. We constructed an item-item similarity matrix using cosine similarity on L2-normalized vectors from the sparse user–item rating matrix. To improve scalability, we preserved only the top-K nearest neighbors for each item, retaining only the top-K neighbors for efficiency. Compared to the baseline models, Item-CF significantly lowered both RMSE and MAE, showing the advantage of modeling item-to-item similarities rather than relying solely on global or mean-based predictions.

Along with rating prediction, we also examined recommendations with ranking metrics at the top-10 level, since users care more about lists than exact scores. We evaluated the recommendation models using Precision, Recall, F1-score, Hit Rate, and Accuracy. User-Based Collaborative Filtering (User-CF) was also tested, which predicts preferences by finding similar users. While this method captures some personalization, its performance was noticeably weaker than Item-CF. User preferences are more diverse and less stable, limiting effectiveness. Item-CF consistently outperformed User-CF across all ranking metrics. For example, about one-third of users received at least one relevant recommendation from Item-CF, compared to only about 4% under User-CF. However, the overall performance remained modest, which can be due to the dataset's extreme sparsity of 98%, where most user–item pairs are unrated.

To further refine Item-CF, we implemented three improvements. First, we applied Top-K similarity filtering to reserve only the strongest similarities, reducing noise. Second, we maintained L2 normalization of similarity values to keep comparisons consistent and reduce popularity bias from highly-rated items. Finally, we kept the recommendations diverse to keep items from being dominated by popular ones to ensure a wider range of suggestions. While these refinements improved efficiency and made the model more robust, they produced only small accuracy improvements, again likely limited by data sparsity. Nevertheless, Item-CF remained the most effective and practical memory-based method in our study because it offers stronger personalization and more actionable recommendations than both baseline models and User-CF.

### **3. Content-Based and Hybrid Approaches**

Building on the foundation of collaborative filtering and baseline models, we further enhanced our recommender system by implementing both content-based and hybrid filtering strategies. Content-based filtering provides a way to recommend movies to users by considering only the attributes of the items themselves, independent of collective user ratings. In our implementation, we constructed movie feature vectors using TF-IDF on titles and release years. For each user, a personalized profile was created by aggregating the features of movies that were rated highly, usually four stars or above. The recommendation process then involved measuring the cosine similarity between this aggregated user profile and all candidate movie vectors, ranking movies by their textual and historical relevance to the user.

This approach offered reasonable accuracy, especially for users or movies with little historical rating data, a scenario known as the cold-start problem. While content-based filtering does not benefit from collaborative signals and thus typically produces lower accuracy than collaborative filtering models, it proved valuable in cases where user-item interactions are sparse, helping ensure every user receives useful suggestions. To address limitations in both content-based and collaborative filtering methods, we developed a hybrid recommender system. This hybrid combined predictions from our SVD-based collaborative filtering model with content-based similarity scores derived from user and movie profiles. Scores were normalized across systems and blended, typically using a weighted average where collaborative filtering was

given greater emphasis. We also pruned the set of candidate recommendations to focus on the strongest matches, ensuring recommendations were diverse, relevant, and less noisy.

The evaluation of both content-based and hybrid models utilized ranking metrics such as Precision at 10, Recall at 10, F1 score, Hit Rate, and Accuracy. Hybrid modeling achieved better robustness than either approach alone, especially benefiting cold-start cases. However, the SVD collaborative filtering model continued to provide the highest hit rate and recall, reinforcing its superiority with this dataset. Nonetheless, the hybrid approach enhanced the system's ability to personalize recommendations and to maintain competitive accuracy, demonstrating the advantages of integrating both behavioral and content signals. In conclusion, the introduction of content-based and hybrid models represented an important advance in our recommender system, addressing the inherent challenges of sparsity and cold-starts while diversifying and improving the recommendations available to users.

#### **4. Results, Limitations, and Future Directions**

After extensive benchmarking and refinement, our analysis shows that advanced collaborative filtering solutions, particularly Item-CF and model-based SVD, outperform naive baselines in rating accuracy and recommendation relevance. Item-based collaborative filtering boosts hit rate and recall by modeling similarities between movies, delivering more targeted suggestions compared to simple averaging approaches. User-CF adds personalization but is less stable, reflecting variability in user behavior.

Model-based methods, such as truncated Singular Value Decomposition (SVD), displayed strong precision and recall, surpassing memory-based approaches in overall performance and especially hit rates for top-10 recommendations. Meanwhile, our refined “Hybrid” approach, which blends collaborative and content-based signals, showed improved balance and robustness, particularly for cold-start scenarios. However, it did not eclipse SVD in absolute performance, affirming latent factor models as the state-of-the-art within our experimental framework.

Despite these successes, several limitations remain. Data sparsity sharply restricts the ability to learn meaningful relationships for both users and many movies, especially those with limited feedback. Cold-start challenges continue to affect new items and a minority of users, and content features like titles and years constrain diversity and novelty. Even with hybridization, richer contextual signals and metadata, deeper models, and context-aware approaches could further elevate accuracy and personalization, while careful tuning of hybrid weights and regularization would likely improve robustness. Looking forward, the project demonstrates that hybrid and model-based algorithms together provide meaningful gains in user engagement and prediction accuracy, but continued innovation is crucial to address real-world challenges like extreme sparsity and cold-starts. Future directions may involve integrating deeper learning frameworks, expanding metadata, and deploying live A/B tests to validate system improvements in actual user environments.