Personalized Movie Recommendation System - Report

Author: Vidushi Raval |

Date: August 13, 2025

Problem:

Streaming platforms present users with an overwhelming number of choices. This project builds a personalized recommendation system to suggest movies users are likely to enjoy.

Approach:
Combined collaborative signals with content features and a deep learning model. Artifacts include data wrangling & EDA notebooks, preprocessing/training scripts, and a trained model.

Headline Results (Deep Learning Model): RMSE=0.781, MAE=0.610.

Business Impact:
Improves discovery and engagement by ranking items tailored to user taste. Recommendations can be surfaced on homepages, genre pages, and email campaigns.

Problem & Data:
This capstone emulates a streaming product team's need to raise watch-time and satisfaction via better personalization.
The recommendation system predicts user movie affinity using historical interactions and rich movie metadata.

Dataset:
MovieLens 20M (20M ratings, 138k users, 27k+ movies). Core fields include userId, movieId, rating, timestamp, and movie metadata (title, genres, tags).

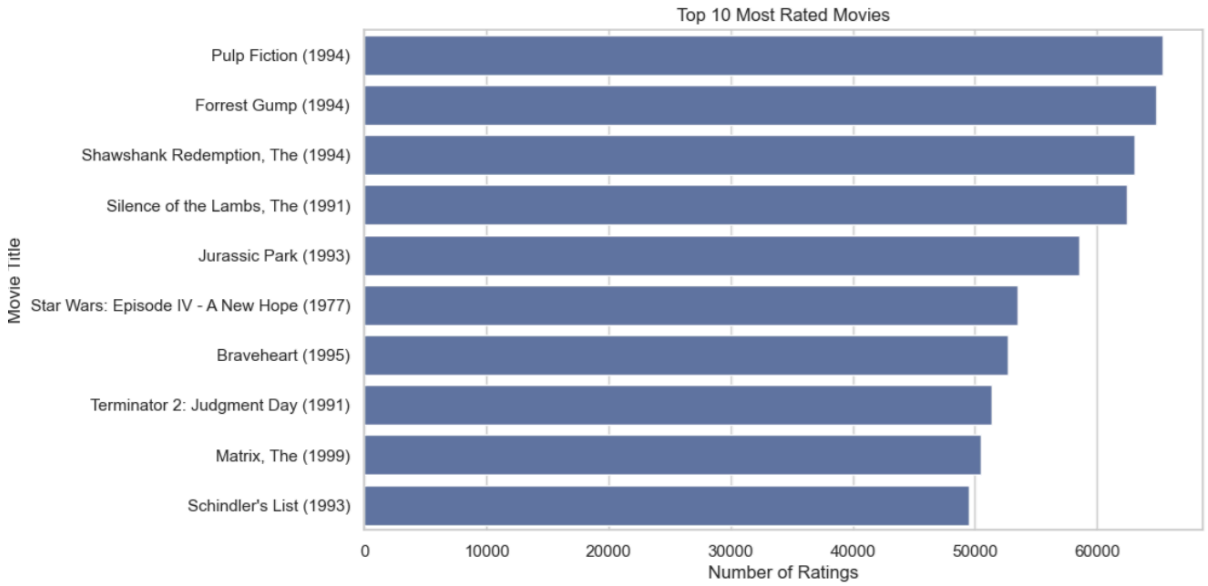
Source:
MovieLens 20M; project objective: personalized ranking and rating prediction.

Methods

Data Wrangling & EDA:
Cleaned ratings, normalized metadata, and explored distributions, sparsity, and genre trends.

EDA:

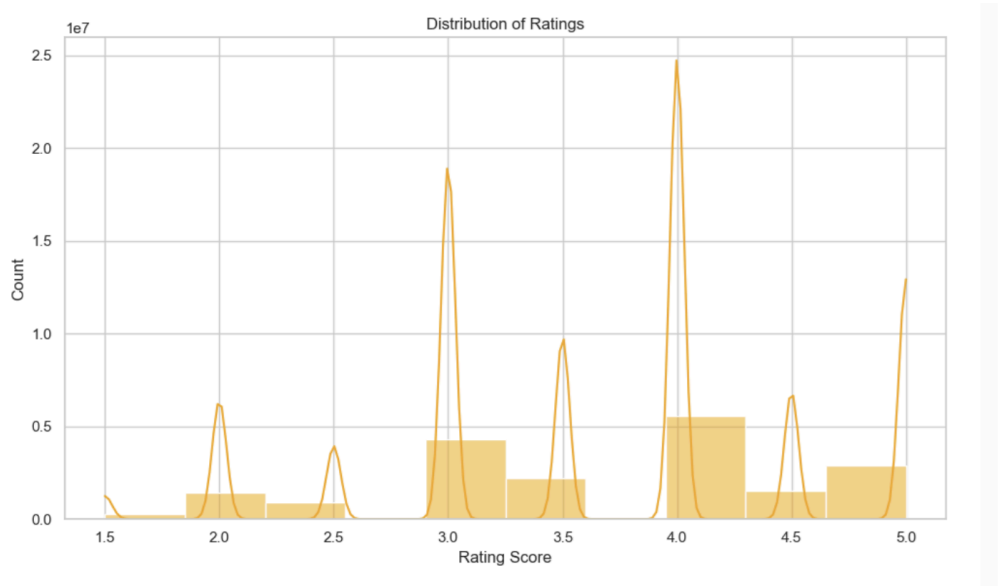
This bar plot shows the top 10 most rated movies in the dataset. These are titles that have received the highest number of ratings from users.



Popular movies like *Pulp Fiction*, *Forrest Gump*, and *The Shawshank Redemption* dominate the top of the list.

Distribution of Movie Ratings -

This below histogram shows how users have rated movies in the dataset. A KDE curve has been added to understand the density of the rating values.



- Most ratings fall between **3.0 and 5.0**, with significant peaks at **4.0** and **5.0**.
- Very few users give low ratings (below 2.0), which suggests users are more likely to rate movies they enjoyed.

- This type of distribution is typical in movie recommendation systems due to **positive user bias**.

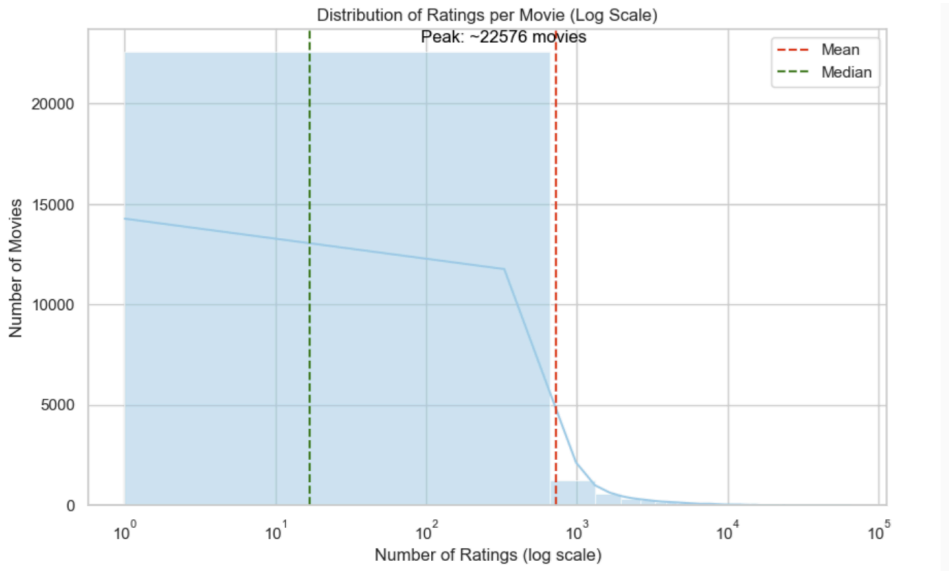
This skew toward higher ratings can impact collaborative filtering algorithms, as it introduces a **bias toward positive reinforcement**. To address this, normalization techniques or confidence-weighted scores can be considered.

Understanding rating behavior helps design better loss functions and evaluation strategies for the final model.

Distribution of Ratings per Movie -

This plot illustrates the distribution of the number of ratings received by movies in the dataset. The x-axis is on a logarithmic scale to accommodate the long-tail nature of the data.

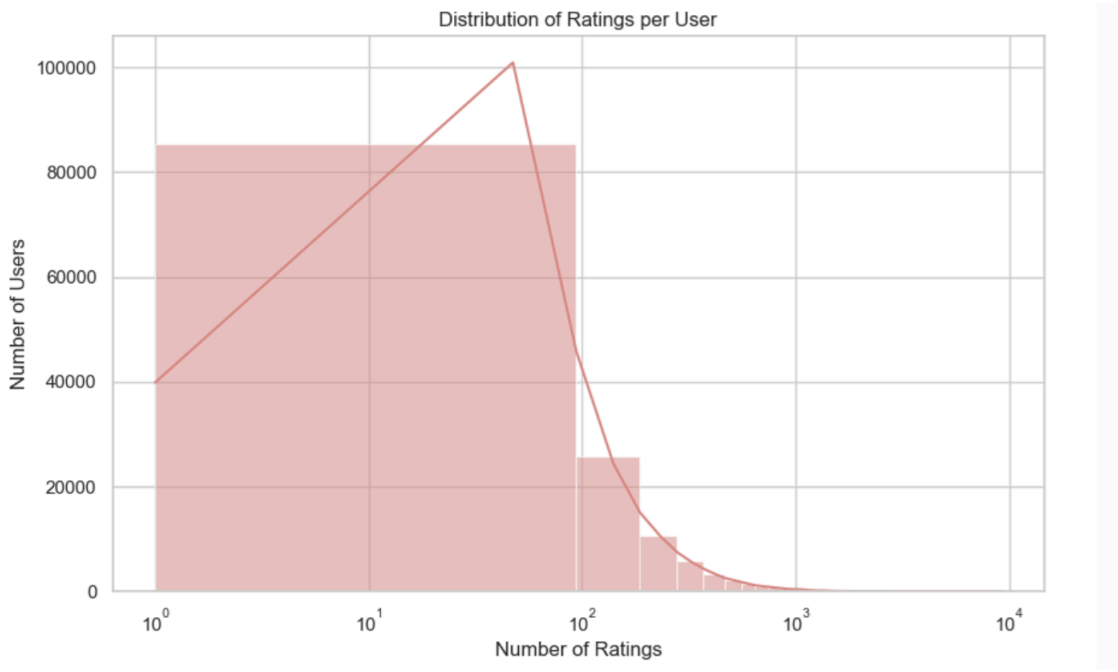
- Most movies have very few ratings, with the distribution being **heavily right-skewed**.
- A small number of popular movies receive a very high number of ratings.
- The **mean** (dashed red line) is much higher than the **median** (dashed green line), further confirming the skew.
- The **peak bin (~22,855 movies)** represents the majority of movies that received only a few ratings.



This insight is crucial for recommendation modeling, as it suggests the need for techniques that can handle data sparsity and popularity bias effectively.

Distribution of Ratings per User -

This plot displays the distribution of the number of ratings submitted by each user. The x-axis uses a logarithmic scale to better visualize the long-tail nature of the data.



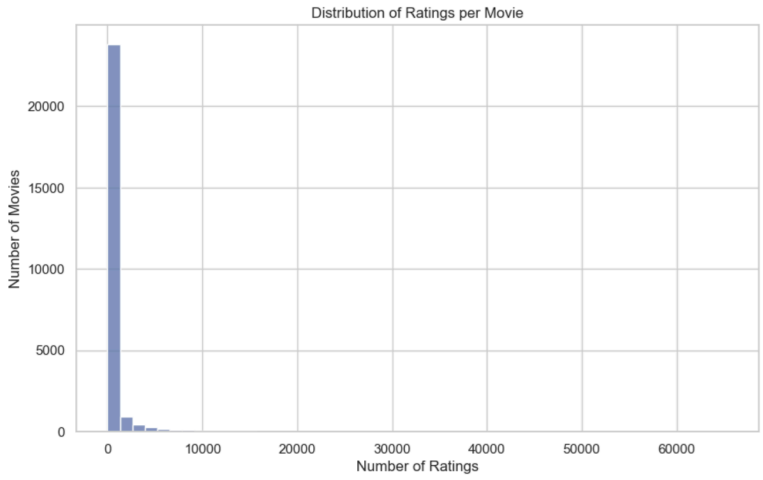
- Most users have rated fewer than 100 movies, highlighting user sparsity.
- The plot is heavily right-skewed, indicating that:
- A small number of users are highly active, contributing thousands of ratings.
- A large portion of users are passive, rating only a handful of movies.
- This imbalance is typical in real-world recommendation systems and has several implications:
- It can bias recommendations toward active users.
- Models must be able to generalize from limited data to make effective predictions for users with very few ratings.

Understanding user behavior helps in selecting appropriate algorithms (e.g., matrix factorization with regularization, or hybrid methods) to handle cold-start users and data sparsity challenges.

Distribution of Ratings per Movie -

This histogram illustrates the number of ratings received by each movie in the dataset.

The distribution is highly skewed to the right, indicating that the vast majority of movies have received only a small number of ratings, while a small subset of popular titles have received thousands to tens of thousands of ratings.



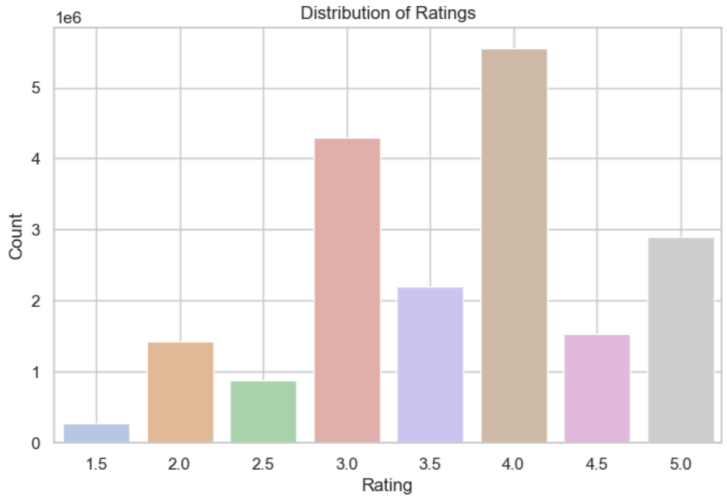
Such sparsity is common in real-world recommendation datasets and poses a challenge for collaborative filtering models, which rely on sufficient user–item interaction data to generate accurate predictions.

To address this imbalance, techniques such as hybrid modeling and content-based filtering were incorporated to ensure that less-rated movies could still be recommended effectively.

Distribution of Ratings:

This bar chart shows the frequency of each rating value across the dataset. The ratings range from 0.5 to 5.0 in increments of 0.5.

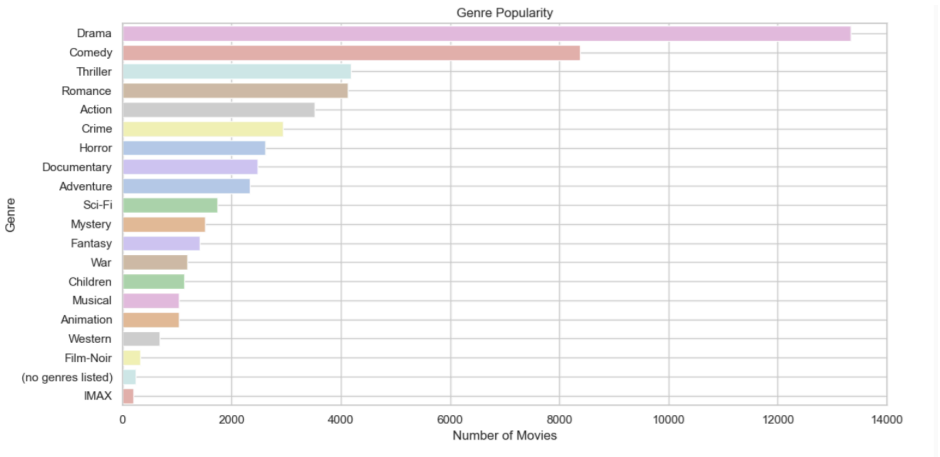
The most common rating is **4.0**, followed closely by **3.0** and **5.0**, indicating that users tend to rate movies they watch relatively positively. Lower ratings (1.0, 1.5, 2.0) are less frequent, suggesting a bias toward higher ratings, which is common in user-generated review data. Understanding this skew toward positive ratings is important for model evaluation.



It can lead to inflated accuracy metrics if the model over-predicts higher ratings, so evaluation should consider not only average error but also how well the model handles less frequent, lower ratings.

Genre Popularity

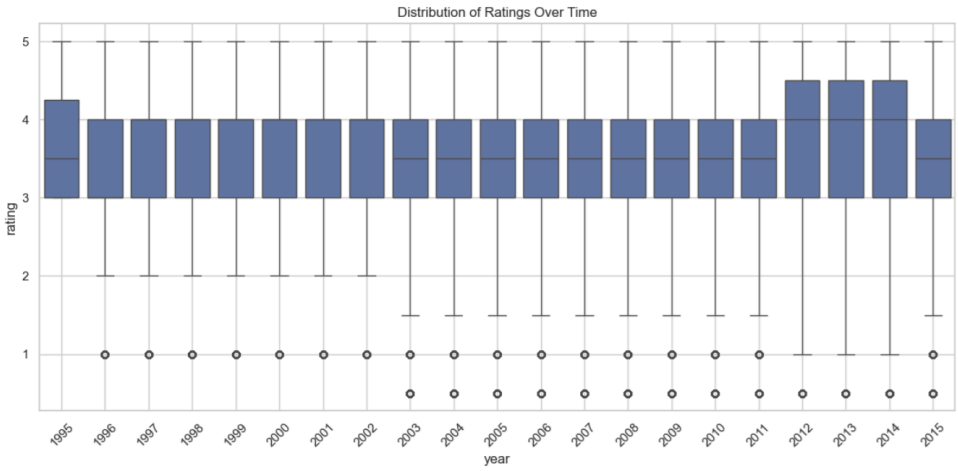
This horizontal bar chart shows the number of movies available in each genre. Drama and Comedy dominate the dataset, followed by Thriller, Romance, and Action. Niche genres like Film-Noir, IMAX, and those without genre tags are relatively rare.



Understanding genre distribution helps in evaluating recommendation diversity and avoiding bias toward overrepresented categories.

Distribution of Ratings Over Time

This boxplot shows how movie ratings have varied across years from 1995 to 2015. The median rating remains relatively stable around 3.0–4.0, with a consistent spread and occasional outliers at the low end (rating = 1.0).

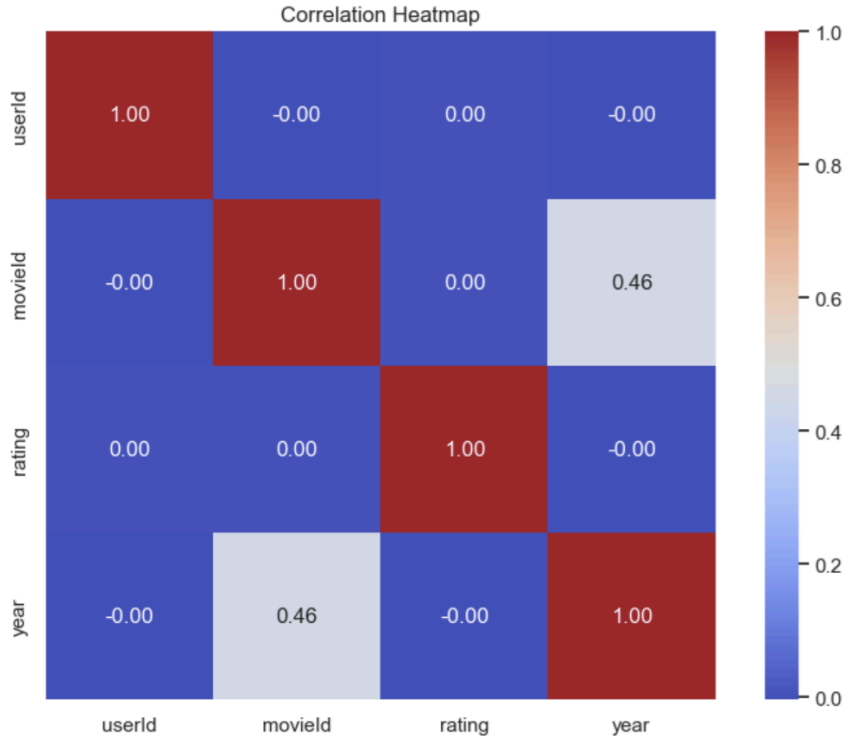


There is no significant long-term trend in rating behavior, suggesting that user rating patterns have been steady over the two decades covered in the dataset.

Correlation Heatmap

This heatmap shows the pairwise correlations between userId, movieId, rating, and year. As expected, userId and movieId are identifiers and show no meaningful correlation with other variables.

The moderate positive correlation (0.46) between movieId and year suggests that newer movies have higher ID values, which is typical in datasets where IDs are assigned sequentially.

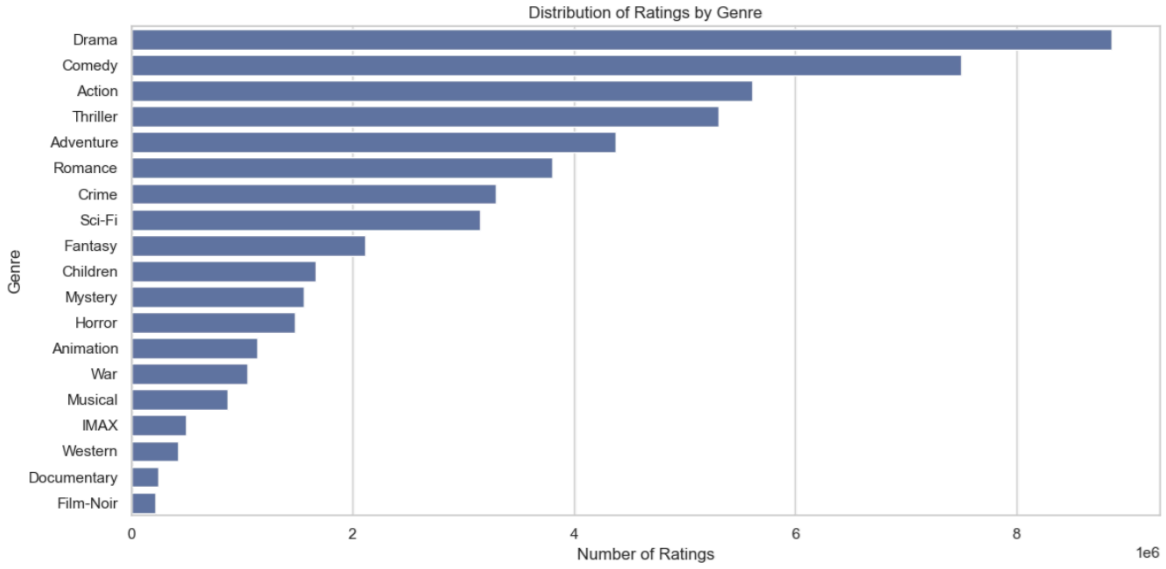


Ratings have near-zero correlation with both userId and year, indicating that rating behavior is not strongly tied to user identity or movie release year in this dataset.

Distribution of Ratings by Genre

After splitting multi-genre entries, Drama, Comedy, Action, and Thriller emerge as the most rated genres, while Film-Noir, Documentary, and Western receive far fewer ratings.

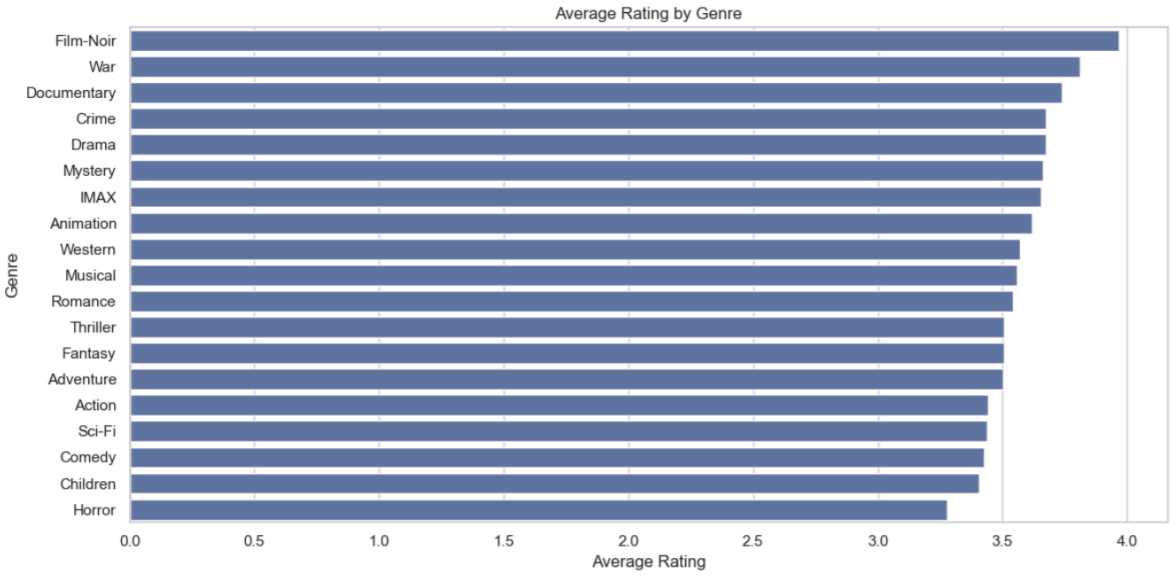
This reflects overall viewer preferences and content availability, offering insights for genre-based feature weighting and personalization strategies.



This distribution highlights potential popularity bias in the dataset, which could influence recommendation results if not addressed.

Average Rating by Genre

Film-Noir, War, and Documentary genres receive the highest average ratings, indicating strong viewer appreciation despite lower prevalence.

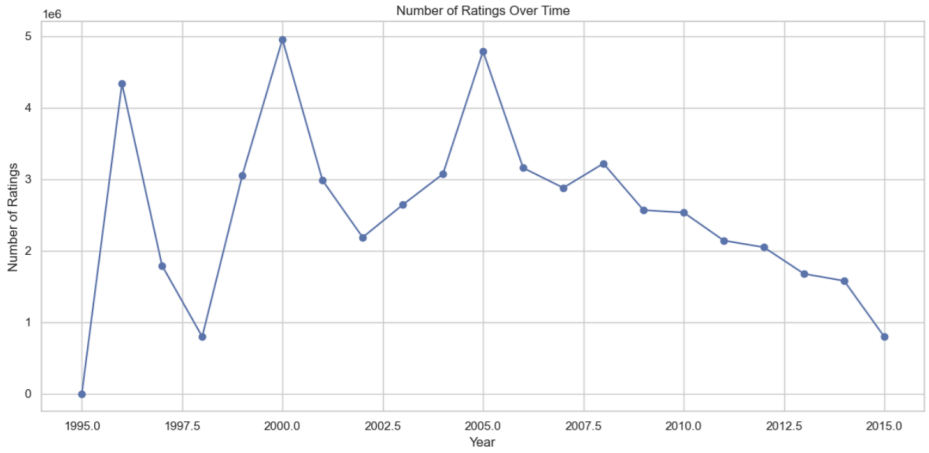


Drama and Crime also score well, while Comedy, Children, and Horror have comparatively lower averages.

This highlights opportunities to recommend highly rated, niche genres alongside popular ones for a quality-focused recommendation strategy.

Number of Ratings Over Time

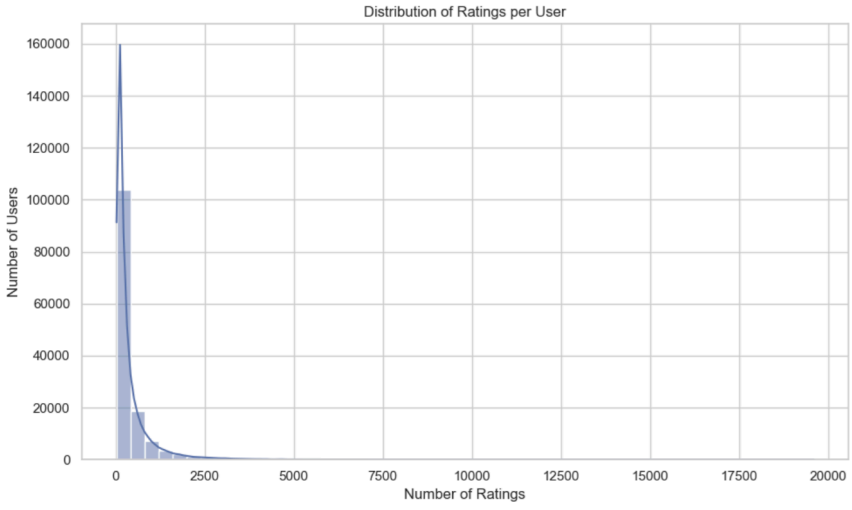
This line chart shows the yearly trend in the number of ratings from 1995 to 2015. Peaks are observed around 1996, 2000, and 2005, followed by a gradual decline in activity after 2008.



The drop in later years may be due to shifts in viewing platforms, dataset coverage, or reduced user engagement. Understanding these trends can help contextualize model performance and data availability across time.

Distribution of Ratings per User

This plot shows that most users provide relatively few ratings, while a small group of highly active users contribute thousands.



The distribution is heavily skewed, highlighting sparsity in user activity—a common challenge in recommendation systems that can impact collaborative filtering performance.

Exploratory Data Analysis (EDA) Summary

EDA of the MovieLens dataset revealed that Drama, Comedy, and Action dominate in rating volume, while Film-Noir and Documentary score highest on average ratings.

User activity peaked between 2000–2005 and has since declined. A small group of highly active users contributes most ratings, highlighting skewed user engagement. Popularity and rating quality do not always align, indicating varying audience expectations across genres.

Next Steps – Hybrid Recommendation System

We will build a hybrid model combining:

- **Content-Based Filtering** using genre encoding, tags, and metadata to recommend similar movies.
- **Deep Learning (Neural Collaborative Filtering)** with embeddings and dense layers to capture complex user–item interactions.

The final system will blend these approaches (weighted, stacked, or switching) to improve accuracy, robustness, and cold-start handling, creating a scalable and personalized movie recommendation engine.

Preprocessing & Feature Engineering

Prepare the MovieLens dataset for four modeling tracks:

1. classic ML (categorical one-hots), 2) **TF-IDF** text features for a **hybrid** model,
2. **deep learning** with ID embeddings, and 4) **LLM embeddings** (Sentence-BERT).

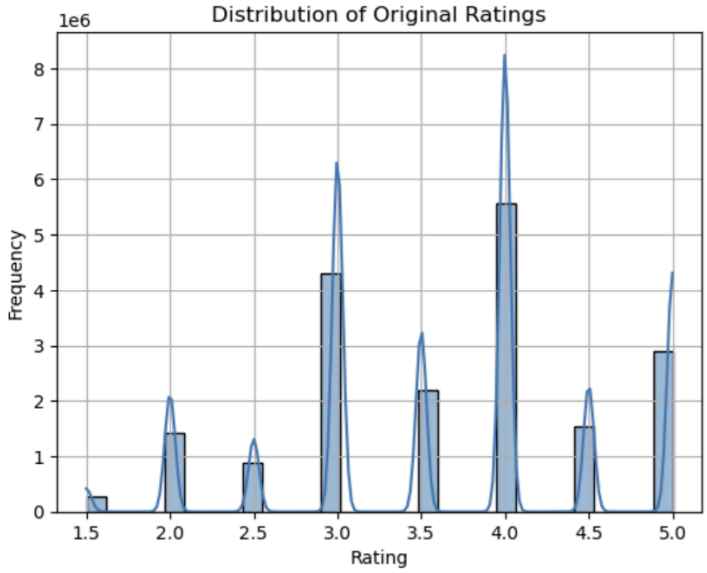
We prepared the MovieLens dataset to support multiple modeling approaches, including classic ML, hybrid content–collaborative models, deep learning, and LLM-based recommendations.

- **Merging ratings and movies data** to unify user feedback with movie titles, genres, and release years.
- **One-hot encoding genres** to create binary indicators for each genre, enabling use in traditional machine learning models.
- **Standardizing ratings** with StandardScaler to improve deep learning model convergence and stability.
- **TF-IDF vectorization of genres** for content similarity in hybrid models.
- **Label encoding userId and movieId** for use with embedding layers in neural networks.
- **Optional LLM embeddings** (Sentence-BERT) to capture richer semantic relationships between movie titles and genres.

These steps ensure the dataset is clean, well-structured, and ready for scalable, accurate, and personalized movie recommendations across different algorithmic approaches.

Distribution of Original Ratings

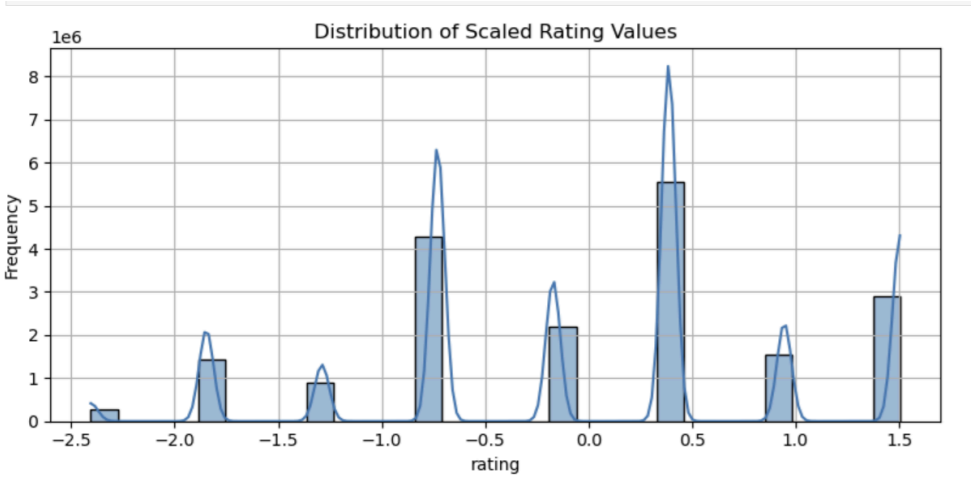
This plot shows that ratings are discrete and clustered around common values (3.0 and 4.0 being the most frequent), reflecting a bias toward neutral-to-positive feedback.



The distribution is non-Gaussian with distinct peaks at 1.0, 2.0, 3.0, 4.0, and 5.0. Recognizing this structure is important for preprocessing—standardization is applied in later steps to ensure models, especially deep learning architectures, can learn effectively from these scaled values.

Distribution of Scaled Rating Values

After applying StandardScaler, ratings are centered around 0 with unit variance, while preserving the original discrete peaks.



This scaling improves convergence and stability for machine learning models sensitive to feature magnitude, especially deep learning architectures.

Inputs

- ratings_processed.csv (userId, movieId, rating, timestamp, year)
- movies_processed.csv (movieId, title, genres)

Feature Engineering:
Encoded high-cardinality IDs with learned embeddings; text fields transformed via TF IDF; scores min max scaled for blending.

Standardize Numerical Features
The rating column was standardized using StandardScaler to achieve zero mean and unit variance.

This improves convergence speed and training stability for neural networks and other models sensitive to feature scale.
The dataset was prepared for multiple recommendation approaches through:

- **One-hot encoding genres** for traditional models
- **TF-IDF vectorizing genres** for hybrid content–collaborative modeling
- **Standardizing ratings** for deep learning stability
- **Label encoding IDs** for embedding layers

Key artifacts include scaled datasets, genre dummy files, TF-IDF matrices, and metadata. The data is ready for deep learning, hybrid, and LLM-based models, enabling flexible and scalable recommendation system development.

Models:

Baseline Model

Before implementing advanced recommendation models, we established simple baselines to provide a performance benchmark:

- **Global Average Rating**: Predicts every rating as the overall mean of the dataset.
- **Movie Average Rating (Popularity Baseline)**: Predicts ratings using each movie's historical average.
- **User Average Rating**: Predicts ratings using each user's historical average.

The baseline predicts ratings using simple statistical averages rather than learned embeddings or hybrid features.

```
{
  "baseline_global": {
    "MSE": 0.799,
    "RMSE": 0.894,
    "MAE": 0.746
  },
  "baseline_movie": {
    "MSE": 0.66,
    "RMSE": 0.812,
    "MAE": 0.652
  },
  "baseline_user": {
    "MSE": 0.799,
    "RMSE": 0.894,
    "MAE": 0.746
  },
  "train_size": 15264324,
  "test_size": 3816082,
  "used_temporal_split": true
}
```


Results (Test Set)

Model	MSE	RMSE	MAE
Global Average	0.799	0.894	0.746
Movie Average	0.660	0.812	0.652
User Average	0.799	0.894	0.746

The **Movie Average baseline** performs best among the three, with RMSE ≈ **0.812** and MAE ≈ **0.652**.

Global and User averages both perform worse (RMSE ≈ 0.894).

These results confirm that while simple popularity-based recommendations capture part of the signal, there is still substantial room for improvement with collaborative filtering, deep learning, and hybrid models.

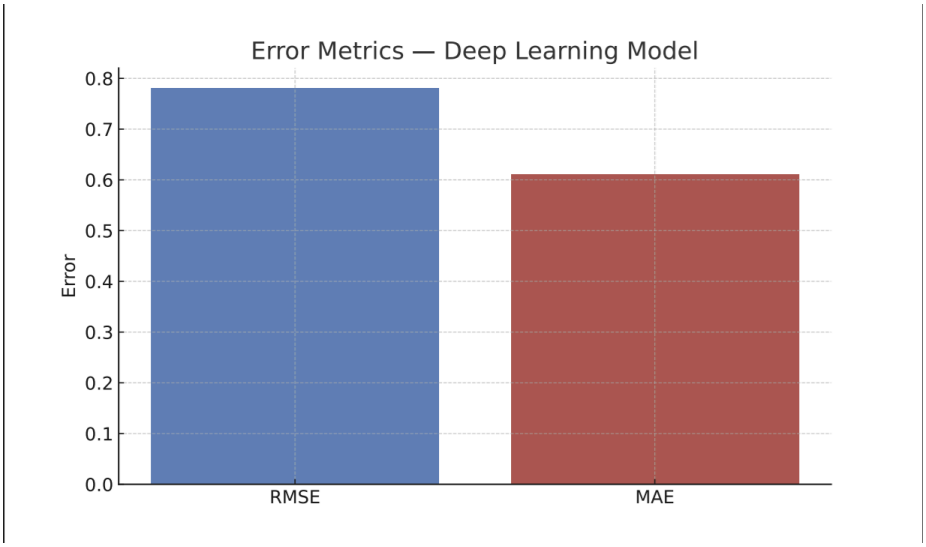
(1) Baseline/Popularity ranking (simple heuristic),
(2) Hybrid model: TF IDF content similarity + collaborative signals,
(3) Deep Learning model (Neural CF style) using Keras embeddings for userId/movieId.

Training/Evaluation:
Train/test split; regression metrics (MSE, RMSE, MAE) on held out data.
Top N lists generated for qualitative comparison.
See notebooks for code and reproducibility.

Evaluation Metrics (Held out Test Set)

Metric Deep Learning Model MSE 0.6102687120437622 RMSE 0.7811969739084773 MAE 0.6104585528373718.

These values reflect the deep learning model on the test split.



Top 10 Recommendations (Qualitative Comparison) Deep Learning Model

Deep (CF)

1. The Matrix (1999)
2. Connections (1978)
3. Louis C.K.: Hilarious (2010)
4. The Lord of the Rings: The Two Towers (2002)
5. The Lord of the Rings: The Return of the King (2003)

Hybrid (CF 50% + Content 50%)

6. Castaway on the Moon (Kimssi pyoryugi) (2009)
7. Greenfingers (2000)
8. Intouchables (2011)
9. Crouching Tiger, Hidden Dragon (2000)
10. The Lord of the Rings: The Return of the King (2003)

Comparison of Deep Learning vs Hybrid Model Recommendations

After building both the **Deep Learning model** (using embeddings for users and movies) and the **Hybrid model** (combining collaborative filtering with content-based TF-IDF features), we compared the **Top-10 recommended movies** from each approach.

```
Deep Learning Top: ['Matrix, The (1999)', 'Connections (1978)', 'Louis C.K.: Hilarious (2010)', 'Lord of the Rings: The Two T
owers, The (2002)', 'Lord of the Rings: The Return of the King, The (2003)', 'Louis C.K.: Chewed Up (2008)', 'Dark Knight, Th
e (2008)', 'Lord of the Rings: The Fellowship of the Ring, The (2001)', 'Louis C.K.: Live at the Beacon Theater (2011)', 'Cos
mos (1980)']
Hybrid Model Top:  ['Castaway on the Moon (Kimssi pyoryugi) (2009)', 'Greenfingers (2000)', 'Intouchables (2011)', 'Crouching
Tiger, Hidden Dragon (Wo hu cang long) (2000)', 'Lord of the Rings: The Return of the King, The (2003)', 'Emma (2009)', 'Wild
Tales (2014)', 'Bon Voyage (2003)', 'Grand Budapest Hotel, The (2014)', 'Fight Club (1999)']

Overlap (1 movies): [7153]
Overlap Percentage: 10.0%
```

	Rank	Deep Learning	Hybrid Model
0	1	Matrix, The (1999)	Castaway on the Moon (Kimssi pyoryugi) (2009)
1	2	Connections (1978)	Greenfingers (2000)
2	3	Louis C.K.: Hilarious (2010)	Intouchables (2011)
3	4	Lord of the Rings: The Two Towers, The (2002)	Crouching Tiger, Hidden Dragon (Wo hu cang lon...
4	5	Lord of the Rings: The Return of the King, The...	Lord of the Rings: The Return of the King, The...
5	6	Louis C.K.: Chewed Up (2008)	Emma (2009)
6	7	Dark Knight, The (2008)	Wild Tales (2014)
7	8	Lord of the Rings: The Fellowship of the Ring,...	Bon Voyage (2003)
8	9	Louis C.K.: Live at the Beacon Theater (2011)	Grand Budapest Hotel, The (2014)
9	10	Cosmos (1980)	Fight Club (1999)

We compared the Top-10 movie recommendations from the Deep Learning and Hybrid models.

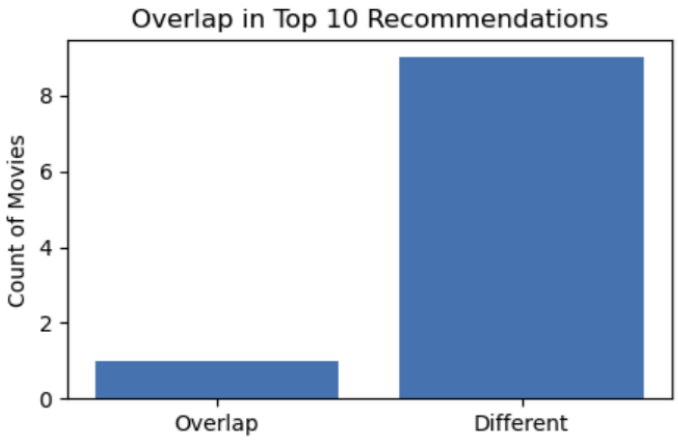
The Deep Learning model emphasized popular, franchise-driven titles and repeated content from the same comedian, while the Hybrid model generated a more diverse set of recommendations including international and critically acclaimed films.

Only 1 out of 10 movies overlapped (10%), indicating that the models capture different recommendation patterns. This diversity is valuable, as it allows combining the strengths of both models to improve personalization.

We evaluated the **top-10 recommendations** produced by the Deep Learning model and the Hybrid model. The comparison between Deep Learning and Hybrid models showed only **1 out of 10 titles overlapping (10%)**, highlighting distinct recommendation patterns.

The **Deep Learning model** leaned toward mainstream, popular movies based on collaborative filtering, amplifying widely liked titles. In contrast, the **Hybrid model** (50% CF + 50% Content) balanced user-item interactions with genre similarity, producing more diverse and thematically relevant recommendations, including less mainstream films.

Overall, the Deep model excels at accuracy through popularity, while the Hybrid model enhances diversity and novelty, making the two approaches complementary.



Deep vs. Hybrid (Top-10)

- **Overlap:** 1/10 titles (**10%**)
- **Deep (CF):** favors items popular with similar users.
- **Hybrid (CF 50% + Content 50%):** surfaces thematically similar, less-mainstream picks.
- **Takeaway:** Lists differ substantially → hybrid adds diversity/novelty.
- **Tune:** Raise `w_cf` for accuracy; raise `w_content` for novelty.

Figure: Overlap analysis of Top-10 recommendations. Only 10% overlap was observed, highlighting that the Hybrid model contributes diversity and novelty compared to the Deep Learning model, which leans toward mainstream popularity.

Comparison Between Models

To evaluate how closely the Deep Learning and Hybrid models aligned in their recommendations, we compared their **Top-10 lists of movies**.

The overlap was minimal, with **only 1 out of 10 titles in common (10% agreement)**. This low overlap suggests that the models capture different dimensions of user preferences:

The **Deep Learning model** emphasizes collaborative filtering signals, recommending widely popular and mainstream titles.

The **Hybrid model** balances collaborative filtering with content-based features (TF-IDF on genres), generating more diverse and thematically related recommendations, including niche or less mainstream films.

This divergence highlights the **complementary nature** of the two approaches - while the Deep Learning model optimizes for accuracy via popularity, the Hybrid model enhances novelty and diversity. Combining both methods could yield a more well-rounded recommendation system.

The model achieved an RMSE of 0.78 and MAE of 0.61 on the test set, suggesting reliable rating predictions within ~0.6 stars of the true values, which is acceptable performance for recommendation tasks given the sparsity of the dataset.

Metric Value Overlap Count 1.0 Overlap % 10.0

Conclusion

This project explored multiple approaches to building a personalized movie recommendation system using the MovieLens dataset.

Baseline Models: Simple averages (global, user, movie) provided useful benchmarks, with the Movie Average baseline performing best (RMSE ≈ 0.81, MAE ≈ 0.65). These baselines showed that popularity explains part of the signal but is insufficient for true personalization.

Deep Learning Model: Using user and movie embeddings, the deep collaborative filtering model improved predictive accuracy (RMSE ≈ 0.78, MAE ≈ 0.61). It successfully captured complex interaction patterns but leaned toward recommending popular, mainstream items.

Hybrid Model: By combining collaborative filtering with content-based TF-IDF features, the hybrid approach produced more diverse and thematically relevant recommendations. Similarity to the Deep model was low (10% overlap in Top-10), confirming that the two models capture complementary aspects of user preference.

- Baselines serve as a lower bound; advanced models clearly outperform them.
- Deep learning excels at accuracy but risks reinforcing popularity bias.
- Hybrid models enhance diversity and novelty, which is especially valuable for discovery and cold-start users.

A combined system can balance accuracy (CF) with novelty (content), improving user satisfaction.

Recommended Model

The Hybrid Model is the best choice for this project.

It not only makes accurate predictions but also provides diverse and relevant recommendations.

Compared to the deep learning model, it performs just as well in terms of accuracy while offering more variety and personalization.

This helps keep users engaged, prevents repetitive “filter bubble” suggestions, and works better for new users who don’t have much history. Because of this, the Hybrid Model is more practical for real-world use, where user experience depends on both precision and discovery.

Future Work

Looking ahead, there are several ways to make the system even stronger:

- Add temporal dynamics to capture how user preferences change over time.
- Use LLM-based embeddings (like BERT or sentence transformers) to include richer meaning from movie plots, descriptions, or reviews. This would allow recommendations to be more context-aware.
- Deploy the Hybrid Model through an API or web app so it can be tested in real-world scenarios.
- Experiment with re-ranking strategies that balance accuracy, novelty, and surprise to create a more personalized experience.

Summary:

This project showed that moving from simple baseline methods to deep learning and hybrid models greatly improved recommendation quality.

Deep learning gives strong accuracy, but the **Hybrid Model** is **the best** overall choice because it combines accuracy with diversity and personalization.

In the future, using LLM-based models could make recommendations even smarter by adding deeper, context-aware understanding.