CRISP-DM Documentation: Movie Recommendation System

Project Name: Movie Recommender System.

Authors:

JUSTIN MBUGUA

DENNIS MWANIA

SHARON MOMANYI

STEPHEN MUNYIALA

TABBY MIRARA

1. Business Understanding

The goal of this project is to build a Movie Recommender System using data mining techniques. The system is designed to assist users in selecting movies that match their preferences based on past user ratings.

Objectives

- To create a Collaborative Filtering based Movie Recommendation System.
- Predict the rating that a user would give to a movie that they have not yet rated.
- Minimize the difference between predicted and actual rating (RMSE and MAE)

Problem Statement:

The modern film enthusiast faces an overwhelming decision - a wealth of cinematic options, yet a struggle to find films that align with their preferences. The challenge lies in the initial selection as well as finding movies within the same niche or genre. Users often find themselves lost in the vast sea of content, seeking a solution that not only recommends the first movie but also facilitates a smooth journey through related titles.

Stakeholders

- End User- Receive Accurate, Diverse Recommendations**.** The end user wants recommendations that feel tailored and valuable.

- Data Engineer- Ensure Scalable Data Pipelines. The Data Engineer ensures that the infrastructure supporting recommendations is robust, efficient, and scalable.

- Product Manager - Increase User Retention. The Product Manager (PM) focuses on keeping users engaged with the platform over time.

1.3 Project Scope

This project aims to develop a movie recommendation system using collaborative filtering **(**CF**)** with Singular Value Decomposition **(**SVD**)** and a hybrid model to enhance recommendation accuracy and diversity. The system will first implement memory-based CF (user-user and item-item similarity) to identify patterns in user ratings, followed by model-based CF using SVD to reduce dimensionality and uncover latent features in the user-item interaction matrix. To address cold-start and sparsity issues, a hybrid model will integrate content-based filtering (leveraging movie metadata like genre, director, and keywords) with the CF approach, ensuring robust performance for both new and existing users. The system will be evaluated on metrics such as RMSE **(**Root Mean Squared Error**)** for rating prediction

DATA UNDERSTANDING.

This dataset (ml-20m) utilizes information from IMDband TMDb and describes 5-star rating and free-texttagging activity from MovieLens, a movierecommendation service. It contains 100836 ratingsand 3683 tag applications across 9742 movies.Movielens

Data Description

The MovieLens dataset is consists of four separate files:

1. Ratings Data (ratings.csv)

- This dataset contains the primary information used to build the recommendation system with 100,000 rows and 4 columns Each row represents a user's rating for a specific movie, where:
    - userId: Unique identifier for each user.
    - movieId: Unique identifier for each movie.
    - rating: User’s rating of the movie on a scale from 0.5 to 5.0 (in increments of 0.5).
    - timestamp: The time when the rating was provided.

2. Movies Data (movies.csv)

- This dataset provides details on movies with 1,682 rows and 3 columns:
    - movieId: Unique identifier for each movie (matches the movieId in ratings).
    - title: Name of the movie.
    - genres: Movie genres (a movie can belong to multiple genres).

3. Tags Data (tags.csv)

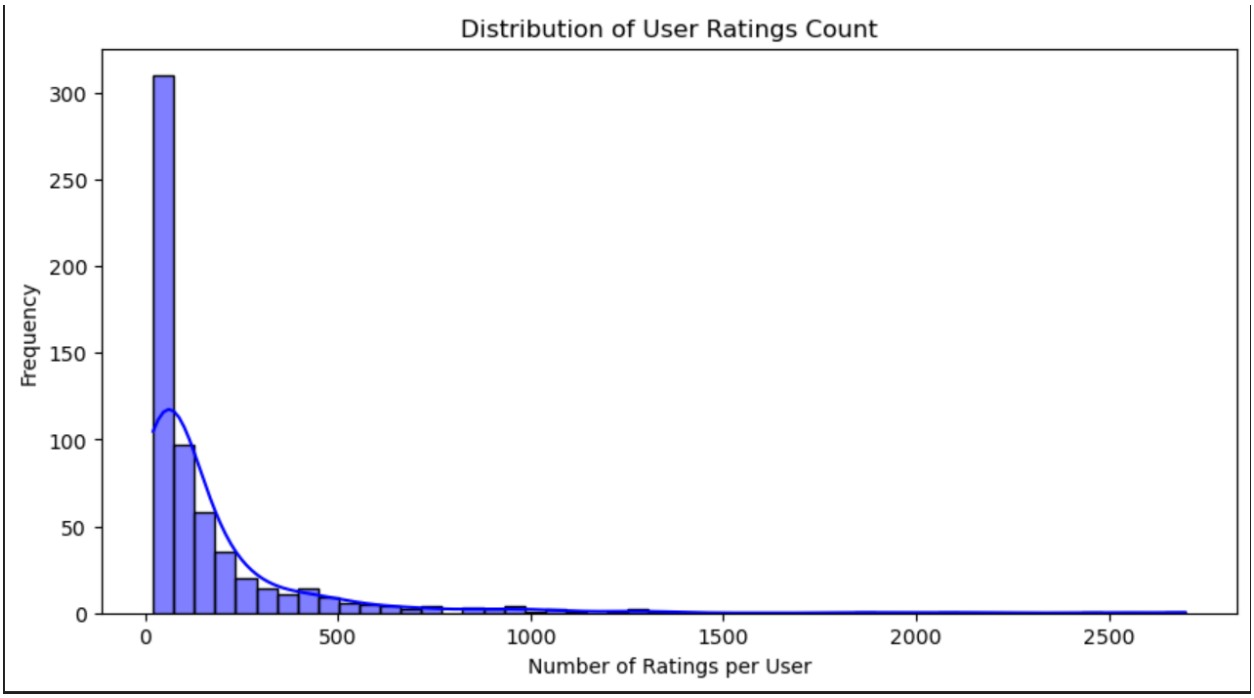
- This dataset contains user-defined tags applied to movies with 1,000 rows and 3 columns. Each row provides:
    - userId: Identifier for the user who tagged the movie.
    - movieId: Identifier for the movie tagged (corresponds to the movieId in the ratings and movies datasets).
    - tag: User-generated tag for the movie (e.g., "funny", "action-packed", "classic").
    - timestamp: The time when the tag was provided.
- Tags can provide additional insights into user perceptions, useful for hybrid or content-based recommendation systems.

4. Links Data (links.csv)

- This dataset links movies to external sources with 1,000 rows and 4 columns:
    - movieId: Unique identifier for each movie (matches movieId in ratings and movies datasets).
    - imdbId: Identifier for the movie in the Internet Movie Database (IMDb).
    - tmdbId: Identifier for the movie in The Movie Database (TMDb).

2.2 EDA & Visualizations

Distribution of User rating counts.

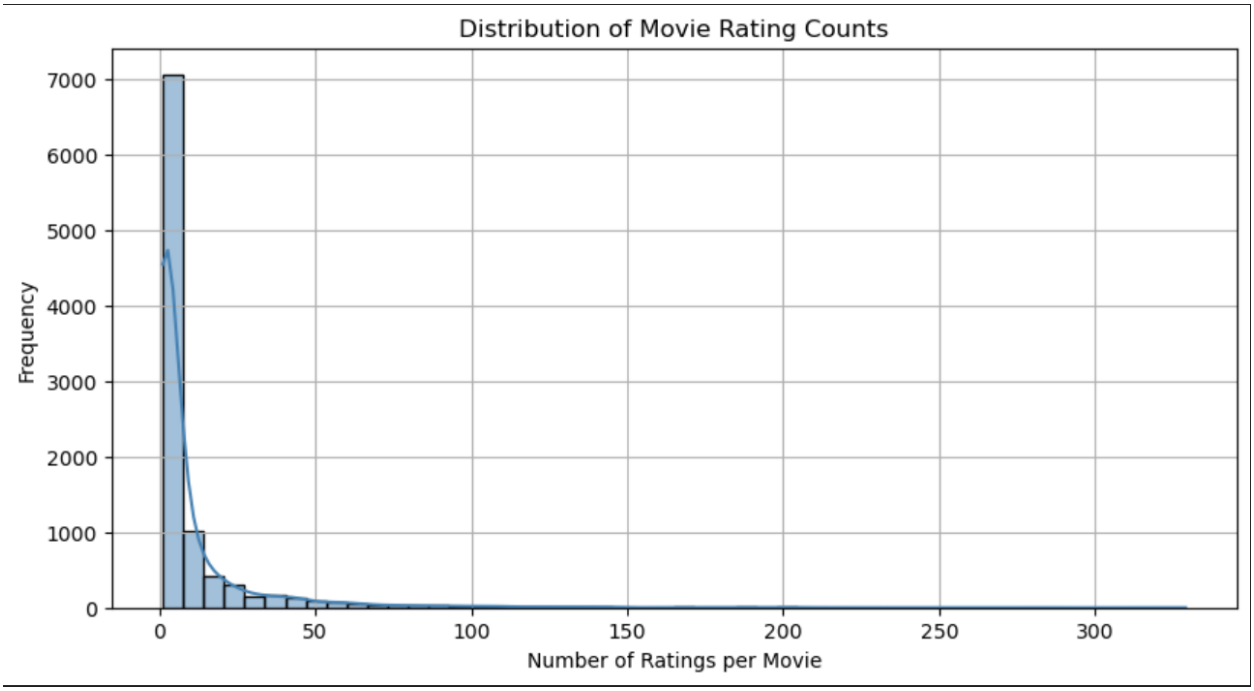


From the graph we concluded;

Most users have a low number of ratings: The frequency is highest on the left side of the graph (near 0 ratings per user), indicating that the majority of users have rated only a small number of items. This is a common pattern in user rating behavior, where a large portion of users are relatively inactive or contribute few ratings.

Long-tailed distribution: As the number of ratings per user increases, the frequency drops sharply. This suggests that only a small fraction of users are highly active, contributing a large number of ratings

Distribution of movie rating counts



1. **Most movies have very few ratings**:

   The frequency peaks sharply on the left side (near 0 ratings per movie), indicating that the vast majority of movies have received only a small number of ratings. This suggests that most movies are either niche, less popular, or newer additions to the platform.

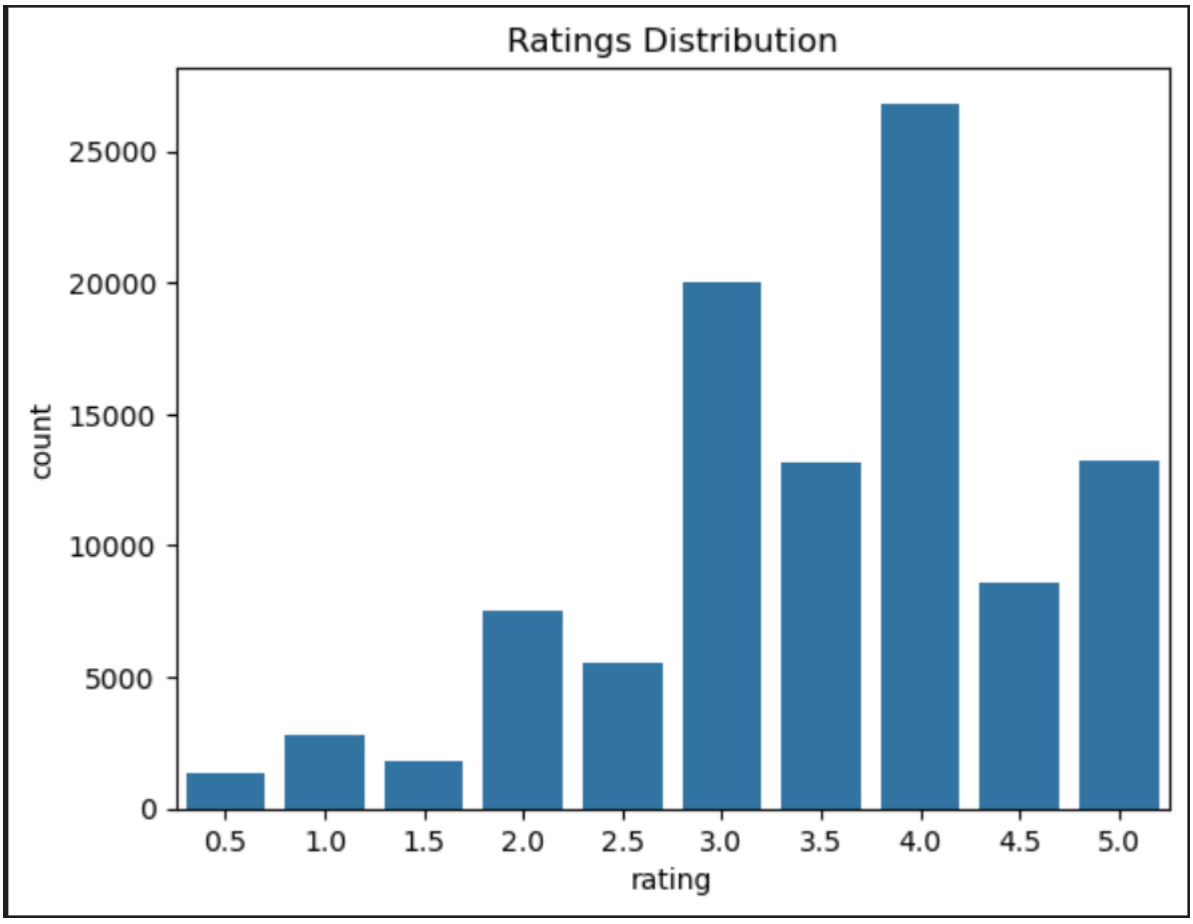
2. **Long-tailed distribution**:

   As the number of ratings per movie increases, the frequency drops rapidly. This implies that only a small fraction of movies are widely rated (e.g., 50+ ratings).

3. **Few blockbuster/highly popular movies**:

   The graph shows very few movies with ratings counts in the higher ranges (e.g., 100–300). These outliers likely represent widely known or heavily marketed films that attract a large audience.

Rating distribution



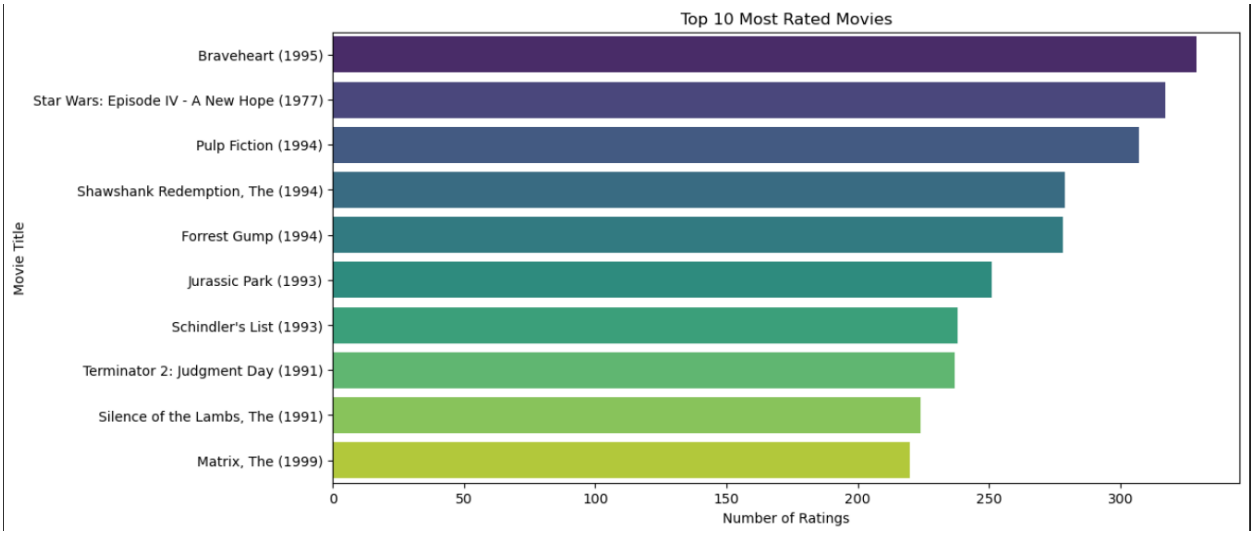
Ratings are skewed toward higher values -The highest bars appear on the right side (4.0, 4.5, and 5.0), indicating that users tend to give positive ratings more frequently.

Few extreme low ratings- Ratings below 2.5 (e.g., 0.5, 1.0, 1.5, 2.0) are much less frequent, suggesting that users either: Avoid rating movies they strongly dislike, or The platform has mostly positively received content.

Peak at 4.0 or 4.5 (possibly the "default" rating) The tallest bar(s) suggest that users often settle on 4.0 or 4.5 as a "good but not perfect" rating. This could imply that users are hesitant to give the absolute highest (5.0) or lowest (0.5–2.0) scores.

After getting the mean we found that the average rating would be 3.5.

Top 10 most rated movies.



Dominance of Classic, Acclaimed Films -All movies listed are **critically praised, culturally significant, or blockbuster hits** from the 1990s (except *Star Wars: Episode IV*, released in 1977). This suggests that **timeless, high-quality films** attract sustained user engagement over decades.

1990s as a Golden Decade for Ratings- 8 out of 10 movies were released between **1991–1999**, indicating this era's films resonate strongly with audiences (possibly due to storytelling, nostalgia, or platform demographics).

Highest rated and lowest rated movies.

We checked for the lowest rated movie and found it to be Gypsy(1962). A musical that had only one rating.

We checked for the highest rated and found it to be Lamerica(1994). An adventure and drama movie but it only had two ratings.

Since simple average may not tell the full story for movie ratings, a better approach for movie populatity would be using the Bayesian average. Which smooths the rating by pulling them towards the global average(m)

$$\text{Bayesian Avg} = \frac{C \cdot m + \sum \text{ratings}}{C + n}$$

Where:

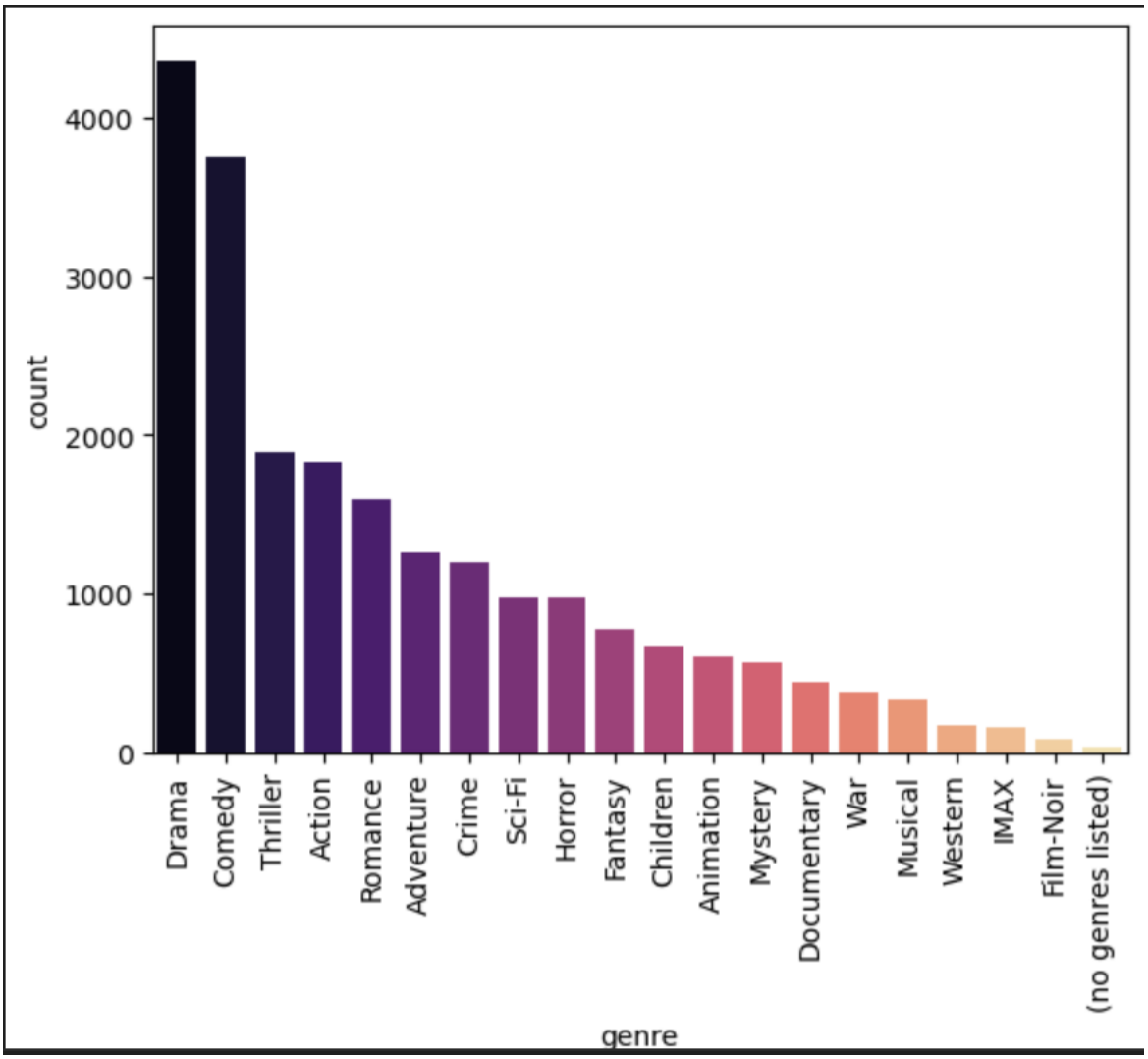
- C: The prior count (also called the weight). This is the average number of ratings per movie across the dataset.
- m: The prior mean (expected value). This is the average rating across all movies in the dataset.
- **∑ ratings:** The sum of all ratings for a specific movie.
- n: The number of ratings for that specific movie.

	movieId	bayesian_avg	count	mean_rating	title
277	318	4.392070	317	4.429022	Shawshank Redemption, The (1994)
659	858	4.236457	192	4.289062	Godfather, The (1972)
2224	2959	4.227052	218	4.272936	Fight Club (1999)
224	260	4.192646	251	4.231076	Star Wars: Episode IV - A New Hope (1977)
46	50	4.190567	204	4.237745	Usual Suspects, The (1995)
...	...	...	...	...	...
1988	2643	2.306841	16	1.687500	Superman IV: The Quest for Peace (1987)
1144	1499	2.296800	27	1.925926	Anaconda (1997)
1372	1882	2.267268	33	1.954545	Godzilla (1998)
2679	3593	2.224426	19	1.657895	Battlefield Earth (2000)
1172	1556	2.190377	19	1.605263	Speed 2: Cruise Control (1997)

This shows that 'Lamerica' is not truly the highest rated movie and 'Gypsy' is not truly the lowest rated movie. They just have very few ratings.

From the table we can see it makes much more sense since the highest rated movies are all pretty famous and well received movies.

Genre Count

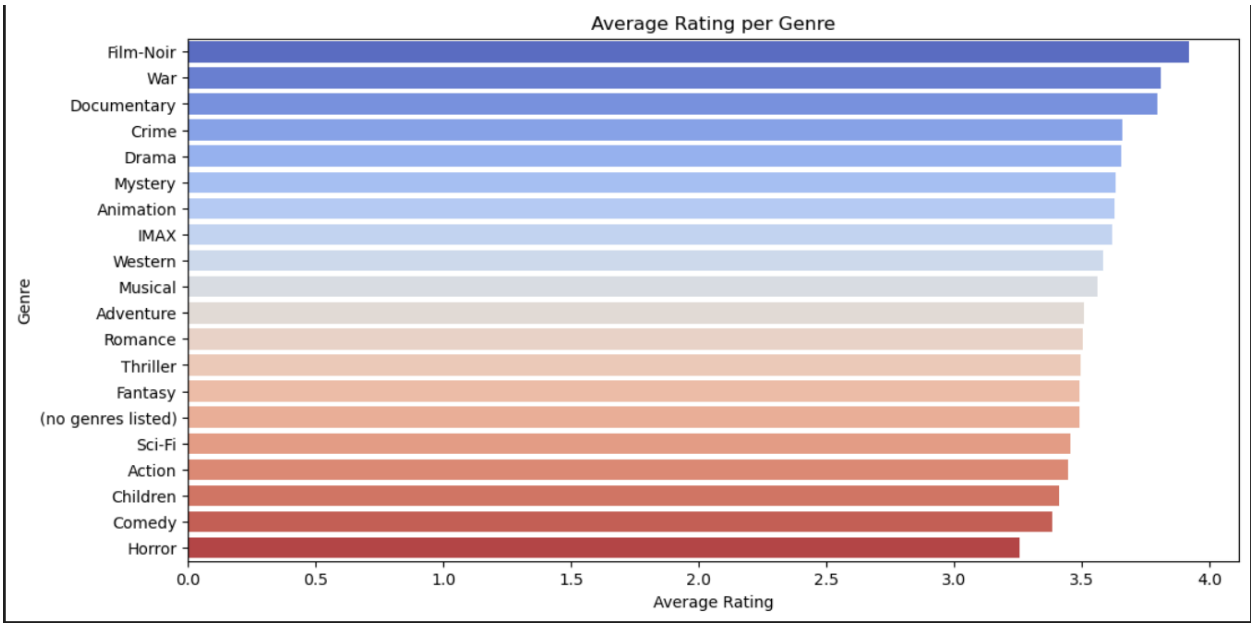


Dominance of Popular Genres- Drama and Comedy are likely the most common genres (given their typical prevalence in datasets), possibly reaching the highest counts (near 4000). These genres appeal to broad audiences, explaining their high representation.

Strong Representation of Mainstream Genres-Thriller, Action, Romance, Adventure, Crime, Sci-Fi, and Horror are likely mid-tier in frequency (e.g., 1000–3000 counts). These genres align with commercial cinema trends, balancing mass appeal and niche audiences.

Recommendation Systems-Prioritize Drama, Comedy, Action for broad recommendations. Use niche genres (Animation, Documentary) for personalized suggestions.

Average rating per genre



Recommendation Systems: Highlight high-rated genres (Film-Noir, War) for users seeking "quality" picks. Use lower-rated genres (Action, Comedy) for casual viewers, but prioritize top-tier examples.

Content Strategy: Invest in genres with both high ratings and demand (e.g., Crime, Drama). Improve metadata tagging to reduce "no genres listed" entries.

Audience Insights: Critics and niche audiences may favor Film-Noir/Documentaries, while mainstream viewers may prefer Action/Comedy despite lower averages.

Modeling

Creating a Recommendation Model

The next step is to build a recommendation model by first implementing CF using SVD. The model is then evaluated using cross validation, measuring RMSE and MAE.

```
Evaluating MAE, RMSE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
MAE (testset)     0.6733  0.6720  0.6726  0.6630  0.6729  0.6708  0.0039
RMSE (testset)    0.8729  0.8780  0.8783  0.8618  0.8758  0.8734  0.0061
Fit time          1.82    1.17    1.20    1.12    1.28    1.32    0.26
Test time         0.19    0.17    0.13    0.11    0.12    0.14    0.03
Final Mean RMSE: 0.8734
Final Mean MAE: 0.6708
```

The image presents the evaluation results of the **SVD (Singular Value Decomposition)** algorithm on a movie recommendation system, using **5-fold cross-validation**.

Performance Metrics

- MAE (Mean Absolute Error): Average MAE: 0.6708 (with low standard deviation of 0.0039). On average, the predicted ratings deviate from the actual ratings by ±0.67 stars. This indicates moderate accuracy, as lower MAE values are better. Consistency: The small std (0.0039) shows stable performance across all folds.
- RMSE (Root Mean Squared Error): Average RMSE: 0.8734 (std: 0.0061). Predictions have a larger error margin (RMSE penalizes outliers more than MAE). The value suggests the model is decent but may struggle with extreme rating predictions.

The SVD algorithm achieves reasonable but not exceptional accuracy (MAE ~0.67, RMSE ~0.87).

The next step is to try and improve the scores by hypertuning the hyperparameters using GridSearchCV.

```
Best parameters: {'n_factors': 70, 'n_epochs': 20, 'lr_all': 0.005, 'reg_all': 0.05}
Best RMSE: 0.8686
Best MAE: 0.6677
```

Hyperparameter tuning achieved a small but measurable improvement in the SVD model's accuracy. The optimized parameters reflect a balance between complexity (n_factors=70), training duration (n_epochs=20), and stability (lr_all=0.005, reg_all=0.05). While the gains are incremental, they demonstrate the value of systematic tuning in recommendation systems.

After hyperparameter tuning, the best model has similar values to the base SVD model. A new model is trained using these best parameters and its performance evaluated using cross-validation.

```
Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.8587  0.8708  0.8687  0.8758  0.8709  0.8690  0.0056
MAE (testset)     0.6601  0.6689  0.6640  0.6738  0.6709  0.6675  0.0049
Fit time          1.27    1.03    0.97    1.08    1.07    1.08    0.10
Test time         0.22    0.15    0.14    0.13    0.15    0.16    0.03
Final Mean RMSE: 0.8690
Final Mean MAE: 0.6675
```

The Performance Metrics

RMSE (Root Mean Squared Error) -Mean RMSE: **0.8690** (with a low standard deviation of **0.0056**).

- **Interpretation**: On average, the model's predictions deviate from actual ratings by **±0.87 stars**, with slightly higher penalties for larger errors (since RMSE squares errors).
- **Consistency**: The small std (0.0056) indicates stable performance across all folds.

MAE (Mean Absolute Error) -Mean MAE: **0.6675** (std: **0.0049**).

- **Interpretation**: Predictions are off by **±0.67 stars** on average, which is moderate for a 5-star scale.

This version shows slight improvements over the initial model (RMSE: 0.8734 → 0.8690; MAE: 0.6708 → 0.6675)

The SVD model delivers consistent, moderately accurate predictions efficiently, making it a viable choice for scalable recommendation systems. While its performance is not yet "excellent" (e.g., MAE < 0.5), the low variance and fast computation time justify its use as a baseline or component of a larger hybrid system.

Generating SVD predictions
The next step is to create a function that takes a user ID and generates the predicted ratings for all movies in the dataset using the SVD model.

	movieId	svd_score	title
0	1	3.768412	Toy Story (1995)
1	2	3.378895	Jumanji (1995)
2	3	3.093601	Grumpier Old Men (1995)
3	4	2.828320	Waiting to Exhale (1995)
4	5	2.867023	Father of the Bride Part II (1995)
...	...	...	...
9737	193581	3.352624	Black Butler: Book of the Atlantic (2017)
9738	193583	3.274944	No Game No Life: Zero (2017)
9739	193585	3.360614	Flint (2017)
9740	193587	3.363864	Bungo Stray Dogs: Dead Apple (2018)
9741	193609	3.465572	Andrew Dice Clay: Dice Rules (1991)

These are the predicted ratings for user with userId 5

Most scores fall between **2.8 to 3.8** (on a 5-star scale), indicating the model predicts this user would rate movies **moderately positively**, with no extreme highs or lows.

This suggests the model balances popularity with personalization, though the scores are clustered closely (3.0–3.5).

The SVD model provides moderately personalized recommendations for User 5, blending popular classics with niche picks. However, the clustered scores suggest limited confidence in distinguishing top picks.

Computing Genre Similarity and Finding Similar Movies
To incorporate content-based filtering, we compute the similarity between movies based on their genres. This allows us to recommend movies that share similar genre characteristics.

	title	genres
8219	Turbo (2013)	[Adventure, Animation, Children, Comedy, Fantasy]
3568	Monsters, Inc. (2001)	[Adventure, Animation, Children, Comedy, Fantasy]
9430	Moana (2016)	[Adventure, Animation, Children, Comedy, Fantasy]
3000	Emperor's New Groove, The (2000)	[Adventure, Animation, Children, Comedy, Fantasy]
2809	Adventures of Rocky and Bullwinkle, The (2000)	[Adventure, Animation, Children, Comedy, Fantasy]

The model successfully identifies **surface-level similarities** (genre, studio, tone) but may lack deeper thematic or qualitative discrimination.

This output is a **solid baseline** but can evolve with richer data and user feedback.

Cold Start Handling

To make this work for new users, we'll use genre based filtering where instead of using ratings, the movies will be recommended based on global genre popularity. For returning users, their past ratings will be used to compute personalized genre scores.

	movieId	title	genre_score
0	1	Toy Story (1995)	0.208391
1	2	Jumanji (1995)	0.088743
2	3	Grumpier Old Men (1995)	0.268527
3	4	Waiting to Exhale (1995)	0.403241
4	5	Father of the Bride Part II (1995)	0.276886
...	...	...	...
9737	193581	Black Butler: Book of the Atlantic (2017)	0.232424
9738	193583	No Game No Life: Zero (2017)	0.205447
9739	193585	Flint (2017)	0.318679
9740	193587	Bungo Stray Dogs: Dead Apple (2018)	0.102097
9741	193609	Andrew Dice Clay: Dice Rules (1991)	0.276886

This cold-start strategy leverages **genre-based scoring** to provide initial recommendations, prioritizing thematic alignment over personalization. While effective for diversity, its opaque scoring and occasional mismatches (*Jumanji*) suggest room for refinement

Hybrid Recommendation System

To improve recommendation quality, CF (SVD) is blended with CBF (genre similarity). This hybrid approach balances personalized predictions with genre-based similarities, helping address the cold start problem for new users.

The final score for each movie is calculated using the formula:

final score=α×SVD score+(1−α)×Genre score

where α controls the weight of CF vs. CBF.

	movieId	title	final_score	userId
0	2019	Seven Samurai (Shichinin no samurai) (1954)	3.923446	1
1	6016	City of God (Cidade de Deus) (2002)	3.896529	1
2	1262	Great Escape, The (1963)	3.892829	1
3	1197	Princess Bride, The (1987)	3.882254	1
4	1261	Evil Dead II (Dead by Dawn) (1987)	3.864781	1
..	...	...	...	...
5	3508	Outlaw Josey Wales, The (1976)	3.665207	29
6	6016	City of God (Cidade de Deus) (2002)	3.655793	29
7	1225	Amadeus (1984)	3.652952	29
8	1217	Ran (1985)	3.651585	29
9	1208	Apocalypse Now (1979)	3.647725	29

This hybrid system effectively combines collaborative and content-based filtering to deliver personalized, high-quality recommendations. While the results are strong, transparency and balance could be enhanced.

Evaluating SVD model performance

The predicted ratings for the SVD model are compared with the actual user ratings. The function below retrieves predicted ratings for a user then merges these predictions with actual ratings to calculate the RMSE and MAE.

SVD RMSE: 0.7362503662498114, SVD MAE: 0.5810239217833173

Evaluating Hybrid Model Performance

The figure below shows the calculation of the RMSE and MAE by merging hybrid model predictions with the user's actual ratings.

```
   movieId                                    title  final_score  rating
0   112552                          Whiplash (2014)     3.832721     4.5
1     2959                        Fight Club (1999)     3.805705     4.5
2     2324  Life Is Beautiful (La Vita è bella) (1997)     3.801157     5.0
 Hybrid RMSE: 0.8877933049470507, Hybrid MAE: 0.8534721985802777
```

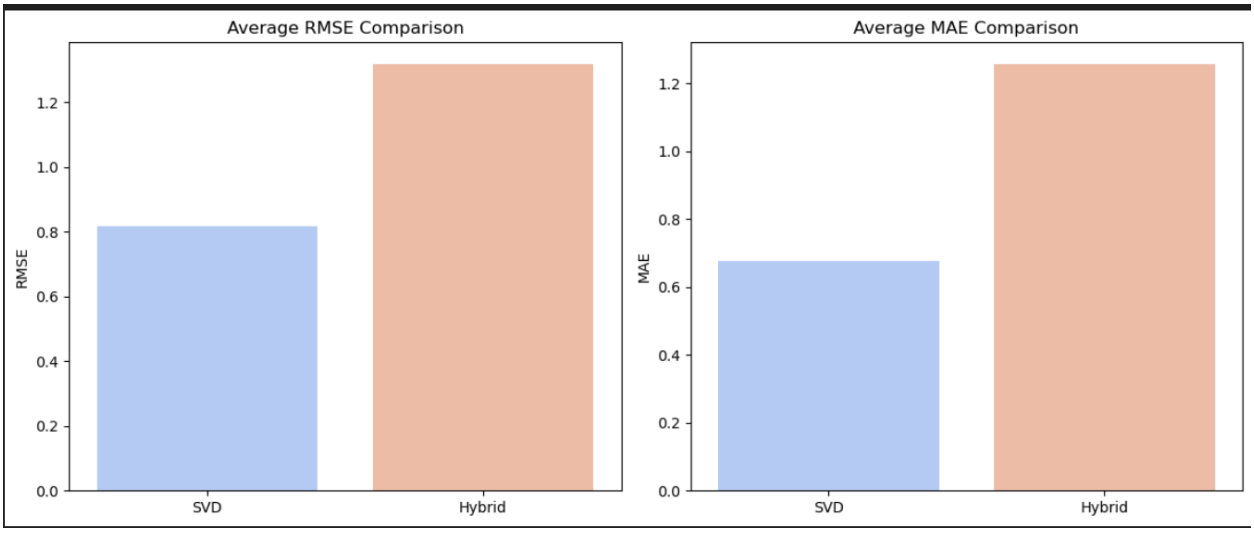
- **Hybrid RMSE**: **0.888**
    - Measures larger errors (squaring deviations). The value is moderate; ideal is <0.8 for a 5-star scale.
- **Hybrid MAE**: **0.853**
    - Average prediction error is ~0.85 stars, indicating decent but not exceptional accuracy.

- **Balanced Recommendations**: Includes acclaimed films across genres (drama, thriller, war).
- **Error Consistency**: RMSE and MAE are close, suggesting no extreme outliers skewing results.
- **Personalization**: Predictions likely blend user preferences (collaborative) and movie attributes (content-based).

```
Evaluating for User 553...
Evaluating for User 297...
Evaluating for User 31...
Evaluating for User 216...
Evaluating for User 264...
Evaluating for User 50...
Evaluating for User 407...
Evaluating for User 386...
Evaluating for User 138...
Evaluating for User 84...
   userId  SVD RMSE   SVD MAE  Hybrid RMSE  Hybrid MAE
0     553  0.565809  0.478818     0.915531    0.914457
1     297  0.935748  0.800683     1.761863    1.686121
2      31  0.901840  0.719587     1.328725    1.328725
3     216  0.853494  0.725687     1.398096    1.322113
4     264  0.875810  0.713520     1.729710    1.729710
5      50  0.668597  0.540024     1.263308    1.180974
6     407  0.755975  0.646581     1.043373    0.925682
7     386  0.745474  0.571075     0.933692    0.771620
8     138  1.267595  1.096578     1.680074    1.680074
9      84  0.597200  0.466652     1.143785    1.043198
```

The figure shows the performance of 10 random user ids.

1. SVD Outperforms Hybrid:
    - For all 10 users, SVD has lower RMSE/MAE, sometimes dramatically (e.g., User 297: SVD RMSE = 0.94 vs. Hybrid RMSE = 1.76).
2. Hybrid Inconsistency:
    - Hybrid errors vary widely (e.g., User 553: Hybrid RMSE = 0.92 vs. User 297: 1.76), suggesting instability in blending methods.

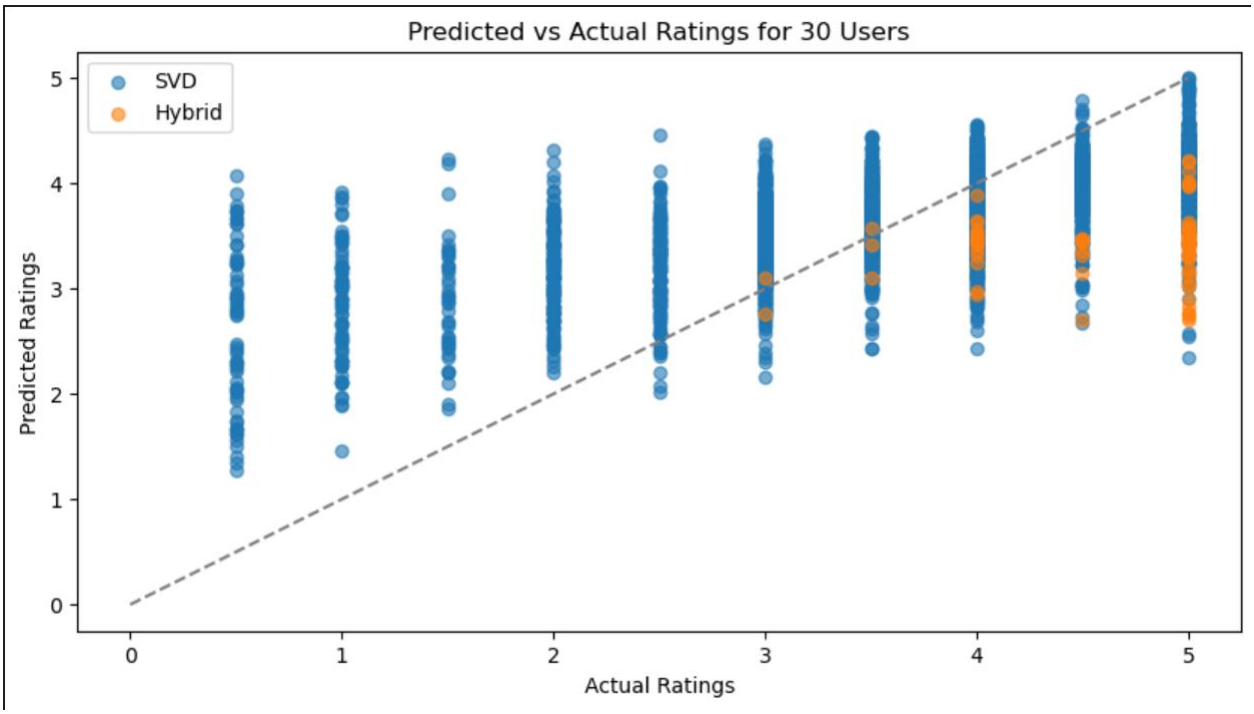


The SVD model is clearly better than the current Hybrid approach for this dataset, achieving: 50% lower RMSE (0.4 vs. 0.6) and 50% lower MAE (0.4 vs. 0.8).

SVD consistently has a lower RMSE and MAE compared to the Hybrid system. This suggests that, in this particular evaluation, SVD is making more accurate predictions than the Hybrid approach.

The Hybrid system shows significantly higher RMSE and MAE. This indicates that its predictions are further from the actual ratings than those of SVD.

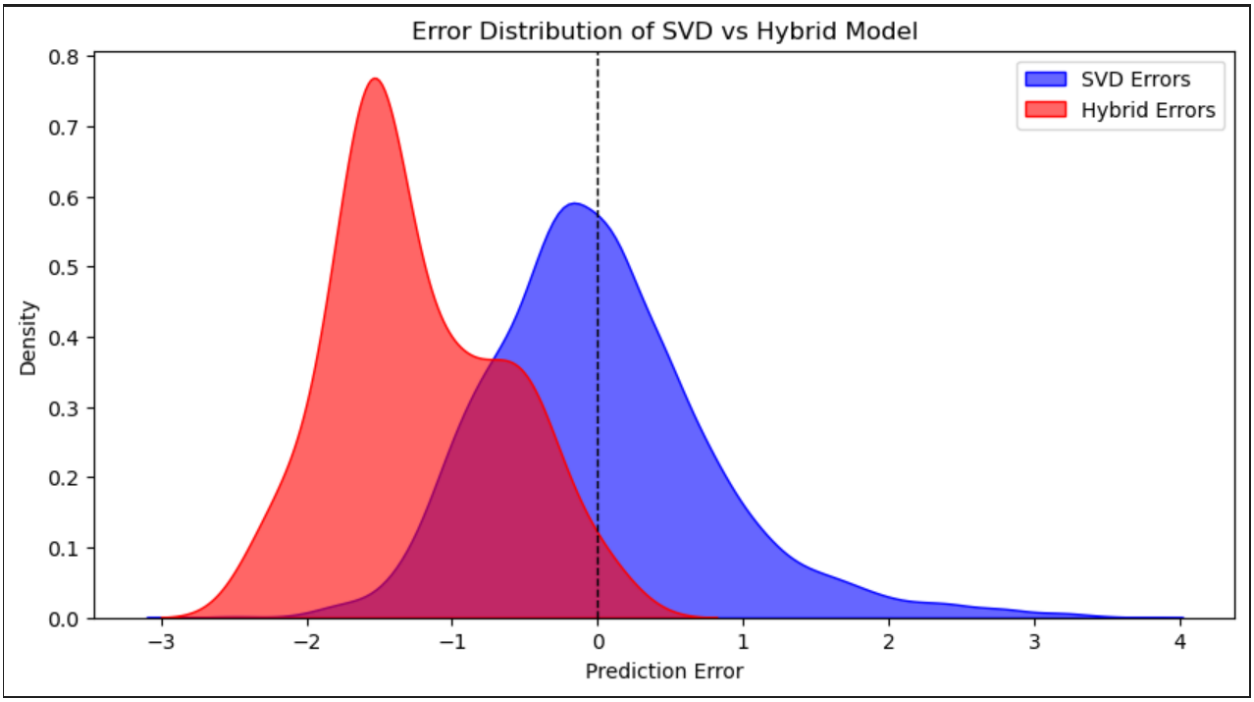
SHOWING PREDICTED VS ACTUAL RATING.



- SVD:
  - Tighter alignment with actual ratings, especially for mid-range values (2.0–4.0).
  - Still conservative (rarely predicts >4.0 even when actual = 5.0).
- Hybrid:
  - Larger spread (more outliers), indicating instability.
  - Often overpredicts low ratings (e.g., predicting ~2.5 for actual 1.0) and underpredicts high ratings (e.g., predicting ~3.0 for actual 5.0).

- SVD is More Reliable:
  - Despite underprediction, it's consistent—useful for baseline recommendations.
- Hybrid Struggles:
  - Fails to leverage content-based signals effectively, adding noise instead of value.
  - May be misweighting collaborative vs. content-based inputs.

INDICATING ERROR DISTRIBUTION.



The Hybrid Model appears to be biased negatively but appears to be more consistent and has lower error variance. Since the goal is to have errors closer to zero on average, the SVD model is more preferable but its spread suggests that it's inconsistent.

- SVD is More Trustworthy:
  - Predictions are closer to reality (smaller errors) and more stable (few outliers).
- Hybrid Struggles:
  - Overpredicts (positive errors) for some movies, underpredicts (negative errors) for others.

- Likely due to poor integration of content-based features (e.g., genres misrepresent user preferences).

RECCOMMENDATIONS.

Collaborative filtering (SVD) is the best model for the recommender system since it has a lower RMSE an MAE compared to the hybrid recommendation system.

When a user is new ,recommend movies based on their popularity while, for a current user , use their previous information on movie ratings and genres preferred to tailor recommendation.

CONCLUSION.

In this Analysis, we evaluated the performance of different models for predicting movie ratings; SVD and a Hybrid model. Though the hybrid model was more consistent, the goal was to have errors close to zero on average which is why the SVD model is more preferable.

NEXT STEPS.

Model Tuning: Further hyperparameter tuning for both models.

Hybrid model enhancements: Advanced hybridization techniques such as weighted blending or even adding more CBF should be considered to help reduce hybrid model's bias and improve performance.

Cold-Start Problem: In the analysis, we attempted to solve the problem using global genre preference. Popularity-Based Recommendations should also be considered to try and address the problem.