Movie Recommendation System

Created by :**Benedict Kuloba**
Allan Kiplagat
Esther Ngaji

Project Overview

In the highly competitive streaming industry, platforms such as Netflix, Amazon Prime, and Hulu strive to retain users and maximize engagement by offering personalized content. Effective recommendation systems are crucial in achieving this goal, helping platforms stand out by tailoring content to individual preferences.

Problem Statement

The primary challenge is enhancing user retention and engagement through personalized movie recommendations. This involves addressing issues such as data sparsity, cold start problems, and scalability while delivering accurate and relevant suggestions to users.

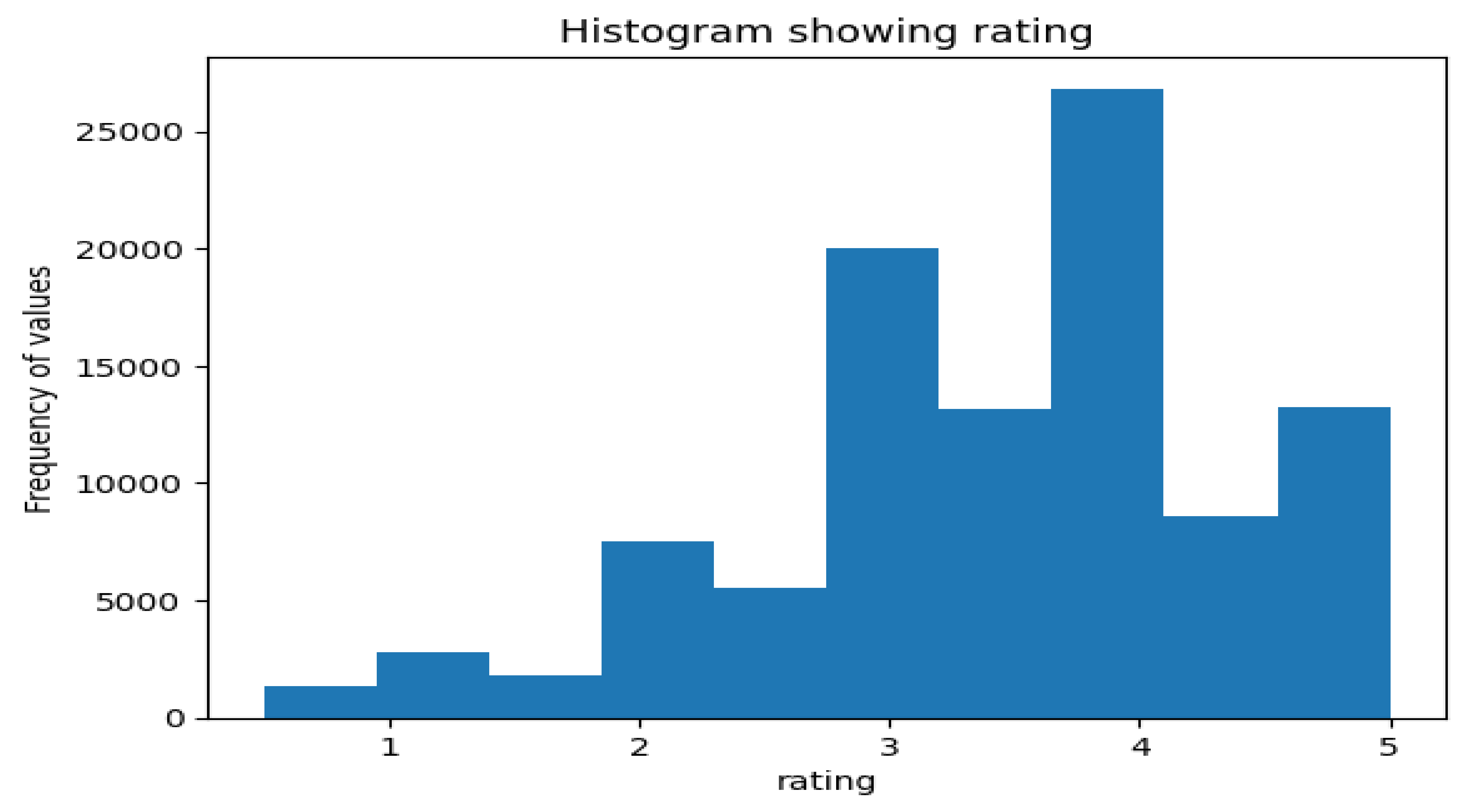
Business Objetives

This project focuses on developing a collaborative filtering-based recommendation system aimed at increasing user engagement and improving user retention for a streaming platform. By leveraging user ratings and preferences, the system will provide personalized movie recommendations that align with each user's unique tastes.

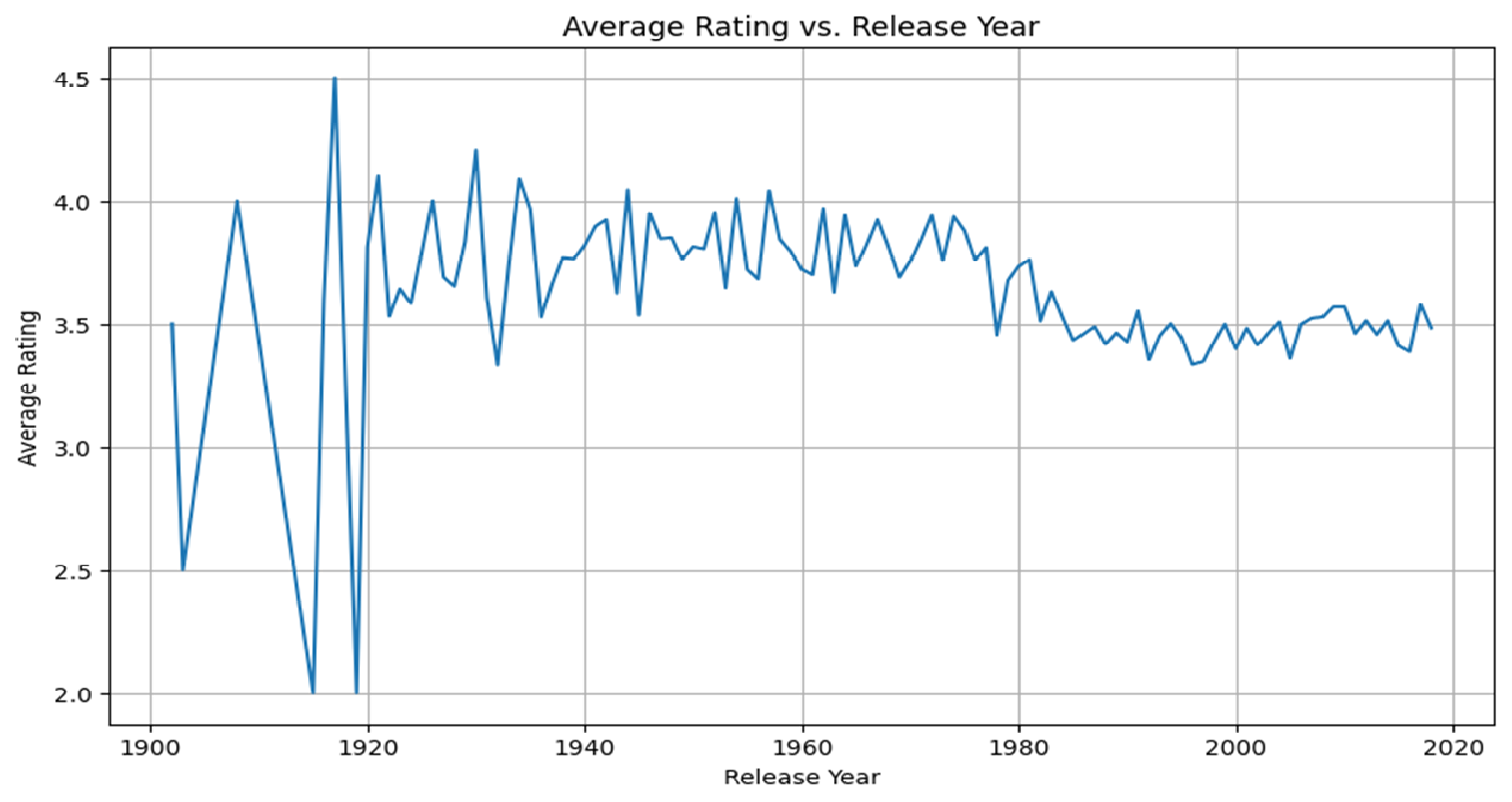
Data Understanding

- The dataset consists of 100,836 ratings and 3,683 tag applications across 9,742 movies. The data were generated by 610 users between March 29, 1996, and September 24, 2018
- Majority of ratings are clustered around the higher end of the scale, particularly around 4.0, indicating that users tend to rate movies positively.
- The mean rating given by users is approximately 3.5. The most common rating in the dataset is 4. Most of the ratings in the dataset are above 3.
- There are fewer low ratings, suggesting that either the movies are generally well-received, or users are more generous with their ratings
- Movies that are more popular (higher number of ratings) in our dataset have higher mean ratings as well.

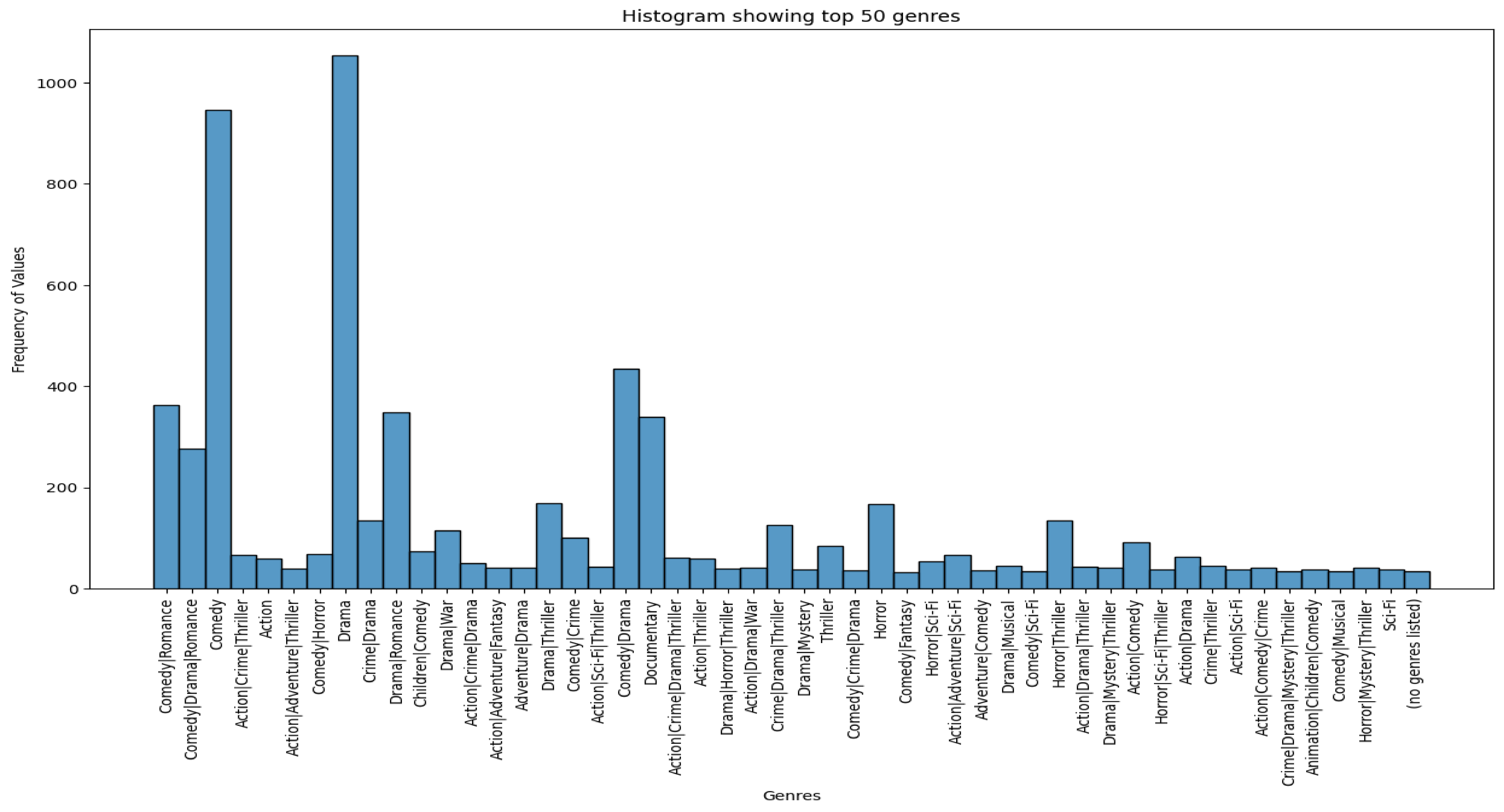
Data Insight



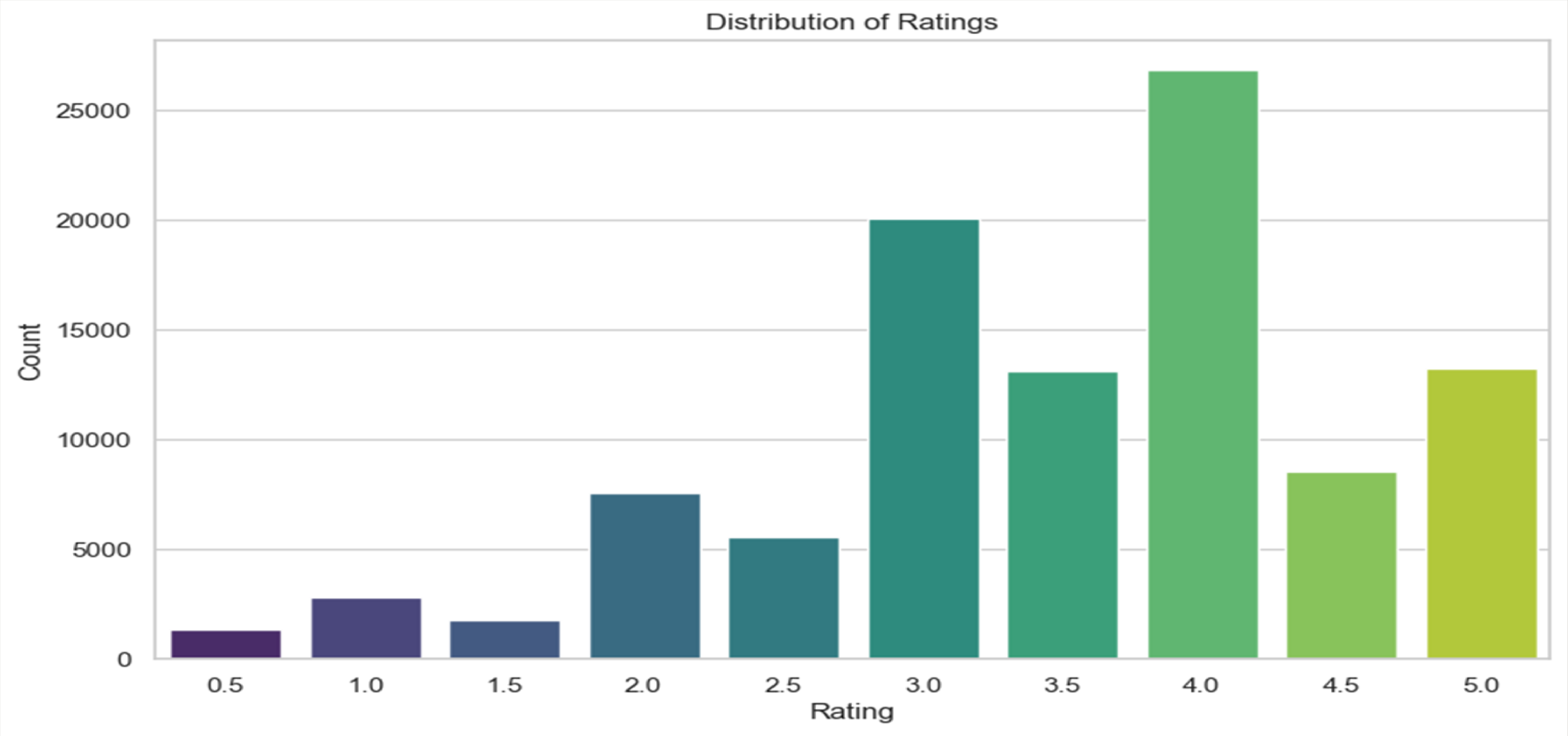
Data Insight



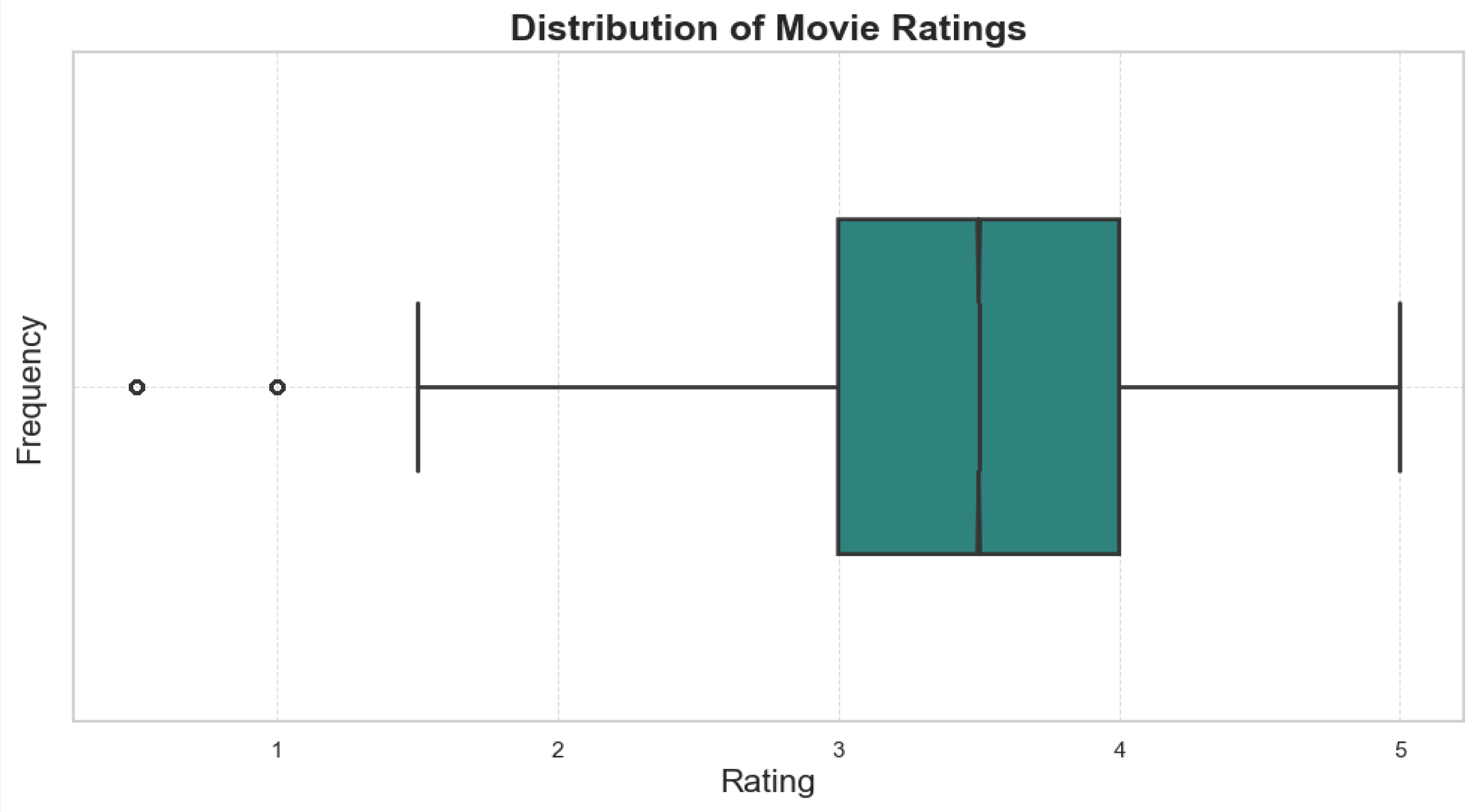
Data Insight



Data Insight



Data Insight



Recommendation System

- Content-Based Filtering: We developed a content-based recommender using movie features like genres.
- This method provides recommendations based on content similarities but may struggle to capture diverse user preferences.
- Neighborhood-Based Collaborative Filtering (KNN): Implemented with SciKit Learn, the KNN model effectively leverages user-item interactions for recommendations.
- The item-based variant, which uses cosine similarity, performed well in identifying similar movies.Model-Based Collaborative Filtering (SVD): Utilizing the Surprise library, we implemented SVD, a matrix factorization technique. SVD achieved a reasonable RMSE of 0.8925, though its effectiveness can be enhanced through hyperparameter tuning.

Recommendation System

- Content-Based Filtering: We developed a content-based recommender using movie features like genres.
- This method provides recommendations based on content similarities but may struggle to capture diverse user preferences.
- Neighborhood-Based Collaborative Filtering (KNN): Implemented with SciKit Learn, the KNN model effectively leverages user-item interactions for recommendations.
- The item-based variant, which uses cosine similarity, performed well in identifying similar movies.Model-Based Collaborative Filtering (SVD): Utilizing the Surprise library, we implemented SVD, a matrix factorization technique. SVD achieved a reasonable RMSE of 0.8925, though its effectiveness can be enhanced through hyperparameter tuning.

Recommendation System

```
In [ ]: # no user-id: (unpersonalized)
final_recommender()
```

```
Unpersonalized recommendation:
```

Out[150]:

	movieId	title	genres	imdbId	tmdbId	userId	rating	timestamp	year_of_release
413	475	In the Name of the Father (1993)	Drama	107207	7984.0	239.0	5.0	1.221159e+09	1993
5302	8784	Garden State (2004)	Comedy|Drama|Romance	333766	401.0	414.0	5.0	1.094519e+09	2004
3832	5378	Star Wars: Episode II - Attack of the Clones (...	Action|Adventure|Sci-Fi|IMAX	121765	1894.0	210.0	5.0	1.473706e+09	2002
581	714	Dead Man (1995)	Drama|Mystery|Western	112817	922.0	122.0	5.0	1.461563e+09	1995
607	762	Striptease (1996)	Comedy|Crime	117765	9879.0	492.0	5.0	8.639761e+08	1996
9518	171749	Death Note: Desu nôto (2006–2007)	(no genres listed)	877057	419787.0	105.0	5.0	1.526207e+09	2007
1923	2551	Dead Ringers (1988)	Drama|Horror|Thriller	94964	9540.0	603.0	5.0	9.539279e+08	1988
691	909	Apartment, The (1960)	Comedy|Drama|Romance	53604	284.0	177.0	5.0	1.435536e+09	1960
709	928	Rebecca (1940)	Drama|Mystery|Romance|Thriller	32976	223.0	59.0	5.0	9.536104e+08	1940
2355	3114	Toy Story 2 (1999)	Adventure|Animation|Children|Comedy|Fantasy	120363	863.0	591.0	5.0	9.705252e+08	1999

Conclusion and Recommendation

- In developing a movie recommender system, we explored three main approaches:
- Content-Based Filtering: Leveraged movie features (e.g., genres). While effective for similar content, it struggled with diverse user preferences.
- Neighborhood-Based Collaborative Filtering (KNN): Utilized SciKit Learn. Item-based variant with cosine similarity performed well in identifying similar movies.
- Model-Based Collaborative Filtering (SVD): Implemented with the Surprise library. Showed reasonable performance (RMSE of 0.8925) but required hyperparameter tuning.

Recommendation

We recommend adopting a hybrid model that combines:

a. **Content-Based Filtering**: To capture content similarities.
b. **Collaborative Filtering**: To leverage detailed user-item interactions.

Additional Recommendations:

- Hyperparameter Tuning: Enhance collaborative filtering performance with further tuning and model evaluation.

- Larger Dataset: Use a more extensive dataset for improved accuracy.

- Regular Updates: Continuously update the recommendation engine based on user feedback and new content.

A well-balanced hybrid model, continually refined and validated, offers the potential for robust and accurate movie recommendations, tailored to specific requirements and user preferences.