

***Table Of Contents***

[BUSINESS UNDERSTANDING 2](#_Toc173347521)

[*Overview* 2](#_Toc173347522)

[*Problem Statement* 2](#_Toc173347523)

[*Objectives* 2](#_Toc173347524)

[*Main Objective* 2](#_Toc173347525)

[*Specific Objectives* 2](#_Toc173347526)

[*Success Metric* 3](#_Toc173347527)

[*Constraints* 3](#_Toc173347528)

[DATA UNDERSTANDING 3](#_Toc173347529)

[DATA PREPARATION 4](#_Toc173347530)

[*Exploratory Data Analysis (EDA)* 4](#_Toc173347531)

[ *Univariate Analysis* 4](#_Toc173347532)

[MODELING 9](#_Toc173347533)

[CONCLUSIONS 10](#_Toc173347534)

[RECOMMENDATIONS 11](#_Toc173347535)

STREAMFLIX MOVIE RECOMMENDATION SYSTEM

# BUSINESS UNDERSTANDING

## Overview

**StreamFlix** is launching a project to develop an advanced movie recommendation system. This hybrid system will blend collaborative filtering with content-based techniques to deliver personalized top 5 movie suggestions based on user ratings, preferences and viewing history. It will also incorporate real-time user interactions and trending content to ensure up-to-date and relevant recommendations. The goals are to enhance user satisfaction, increase watch time and boost customer retention. By offering superior content curation and optimizing content strategies, StreamFlix aims to stand out in the competitive streaming market and strengthen its position as a leader.

## Problem Statement

Streamflix is experiencing major issues such as high churn rates and a significant number of users canceling their subscriptions within the first three months. Engagement has decreased as the average watch time per user has declined by 15% over the past year. Users are also struggling with content discovery spending an average of 20 minutes per session browsing before selecting a movie which leads to frustration and reduced satisfaction. Increasing competition from rival streaming services which offer more personalized user experiences is further impacting Streamflix's market share.

## Objectives

### Main Objective

To develop and deploy a hybrid recommendation system that combines collaborative filtering and content-based filtering techniques to enhance recommendation accuracy by capturing both user-item interactions and movie attributes ensuring a more personalized movie recommendation experience.

### Specific Objectives

1. To build a collaborative filtering model using user ratings to generate top 5 movie recommendations, leveraging algorithms such as Singular Value Decomposition (SVD) and k-Nearest Neighbors (k-NN).
2. To address the cold start problem for new users by integrating content-based filtering, utilizing features such as movie genres, directors, and cast.
3. To evaluate the hybrid recommendation system using appropriate metrics like Root Mean Square Error (RMSE), Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG) to ensure accuracy and relevance of the recommendations.

## Success Metric

* Root Mean Square Error (RMSE) < 0.9 for rating predictions

## Constraints

1. Data Limitations: Small dataset size leading to potential bias due to demographic skew and time period of ratings and lack of additional movie metadata beyond the information provided in the datasets.
2. Data Sparsity: Many users may have rated only a small fraction of available movies creating challenges in finding similar users or items with limited data points.
3. Cold Start Problem: Difficulty providing accurate recommendations for new users or newly added movies, limited effectiveness of collaborative filtering for users with few ratings.
4. A/B Testing Capabilities: Limitations in conducting extensive A/B tests to compare different recommendation algorithms.
5. Ethical Considerations: Ensuring fairness in recommendations across different user groups and balancing business objectives with ethical recommendation practices.
6. Genre and Diversity: Ensuring a balance between accuracy and diversity in recommendations and avoiding over-specialization in user recommendations.
7. Privacy and Data Protection: Ensuring user data privacy and compliance with regulations like GDPR that establishes the general obligations of data controllers and of those processing personal data on their behalf (processors).
8. Evaluation Metrics: Limitations of evaluation metrics in reflecting real-world user satisfaction and a lack of direct user feedback on recommendation quality.

# DATA UNDERSTANDING

The data used in this project is the Movielens dataset from GroupLens Research Lab which includes movie ratings from 1902 to 2018. This dataset comprises 100,836 ratings and 3,683 tag applications across 9,742 movies, with each user rating at least 20 films. Although the full dataset includes 1.9 million ratings, we focused on a subset of approximately 100,000 ratings for our current model due to time and resource limitations. This sample size provides a balance between computational efficiency and statistical relevance for developing our recommendation engine. The datasets involved are links.csv, movies.csv, ratings.csv, and tags.csv. For our recommendation system, we utilized the features movieId, userId\_x, rating, title and genres. After merging the datasets on the movieId column, we obtained a DataFrame with 285,783 rows and 11 columns. While this data sufficed for our objectives, additional information such as actors, directors, production studios, runtime and user demographics could have offered more context and insights into user preferences and movie characteristics potentially enhancing the recommendations.

# DATA PREPARATION

To prepare the data for our analysis, we started by creating a class specifically designed to explore the datasets. We decided to merge all the datasets in order to get a holistic view of the data we would be working with. The first step was to drop columns we deemed least relevant to our model curation process.

The second step was to check the integrity of the data and we found that there were 21 missing values in the merged dataset. We opted to drop the rows with the missing values as we would still have sufficient data for our analysis.

We then examined the properties of the DataFrame columns. Initially, we noticed that the user\_id column was of type float. Given that user IDs are inherently integer values, we converted this column from float to integer to ensure data consistency.

After these adjustments, we used the data\_explorer.merged\_data.info() function to review the structure of our DataFrame. It revealed that we had 100,818 entries and six columns: movieId, title, genres, user\_id, rating, and release\_year. Each column contained non-null values with user\_id having been successfully converted to an integer data type. With these preparations completed, we were ready to proceed to exploratory data analysis confident that our dataset was now well-structured and suitable for further investigation.

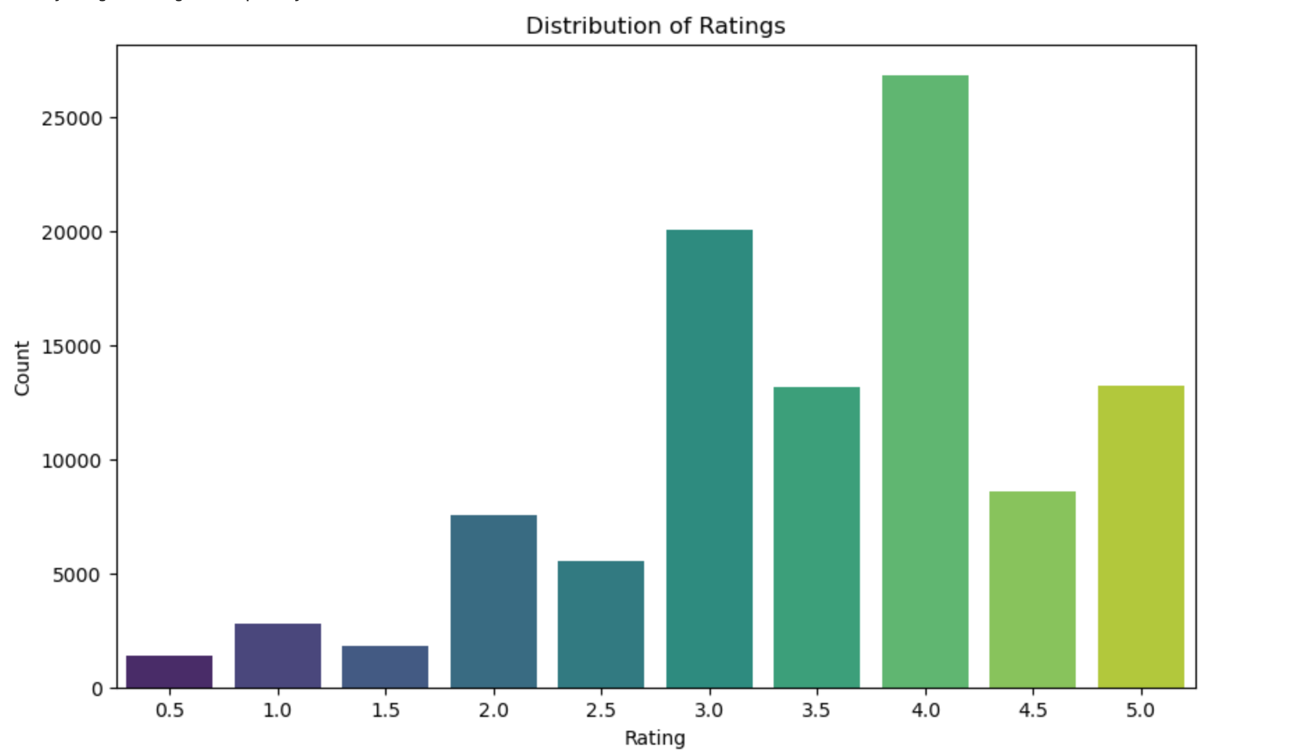
## Exploratory Data Analysis (EDA)

For our exploratory data analysis (EDA), we began by analyzing various aspects of the dataset to uncover patterns and insights.

### Univariate Analysis

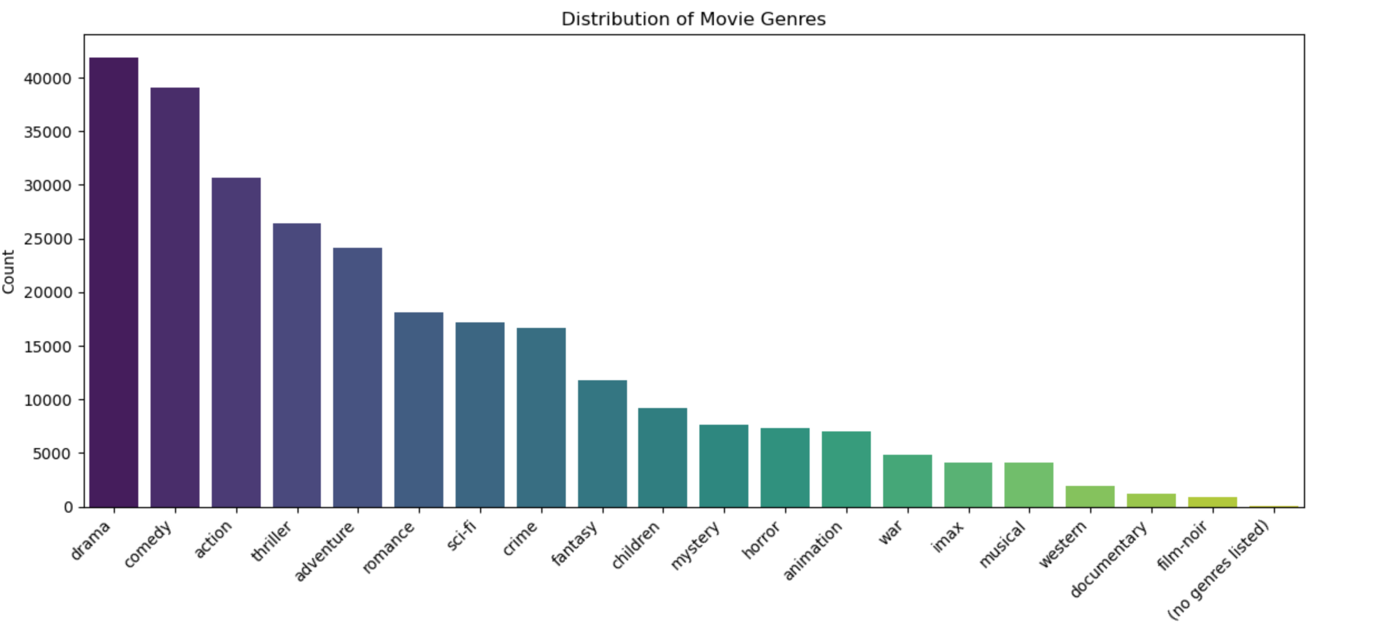
In our univariate analysis, we examined several key aspects of the dataset to gain deeper insights.

1. Distribution of ratings



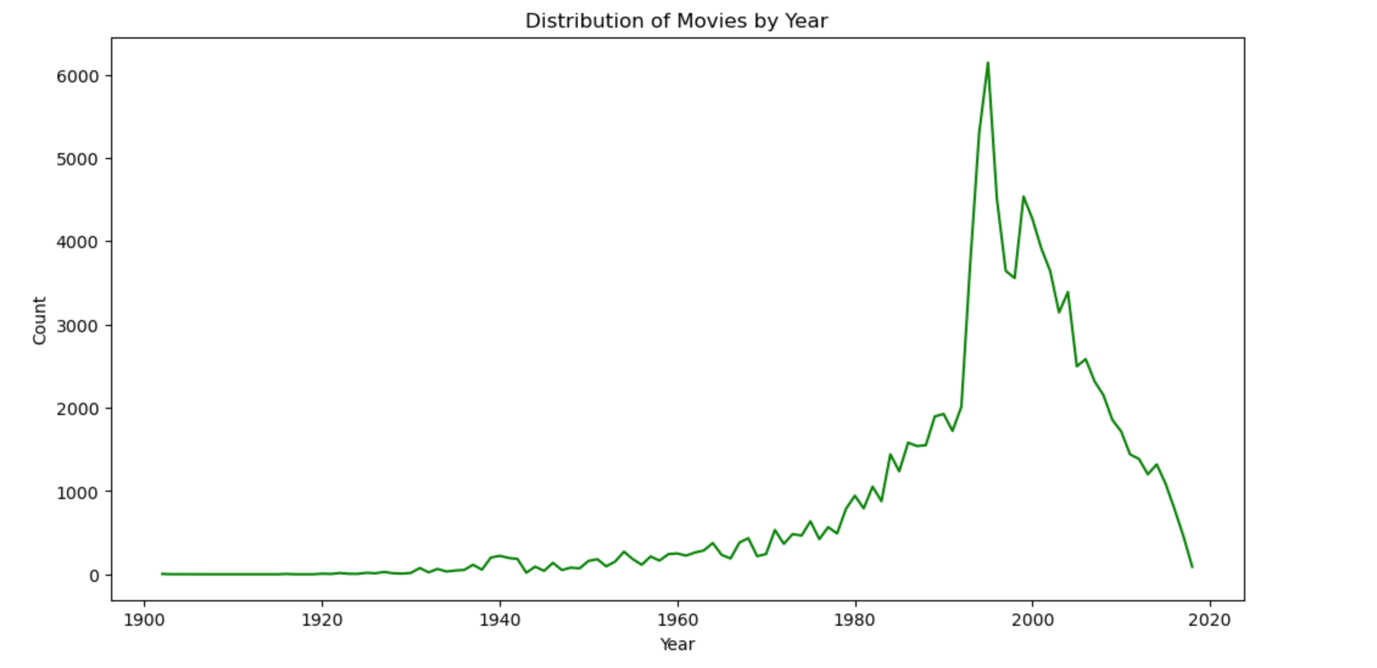
We began with the “Distribution Of Ratings” plot which revealed that the most common rating was 4.0 with around 25,000 movies receiving this score. The distribution was positively skewed indicating a tendency towards higher ratings. Significant counts were also observed at ratings of 3.0 and 3.5. Although ratings of 2.0, 4.5 and 5.0 were less frequent, they were still notable. Extremely low ratings of 0.5 and 1.0 were rare suggesting that most movies were generally rated moderately well.

1. Distribution of genres



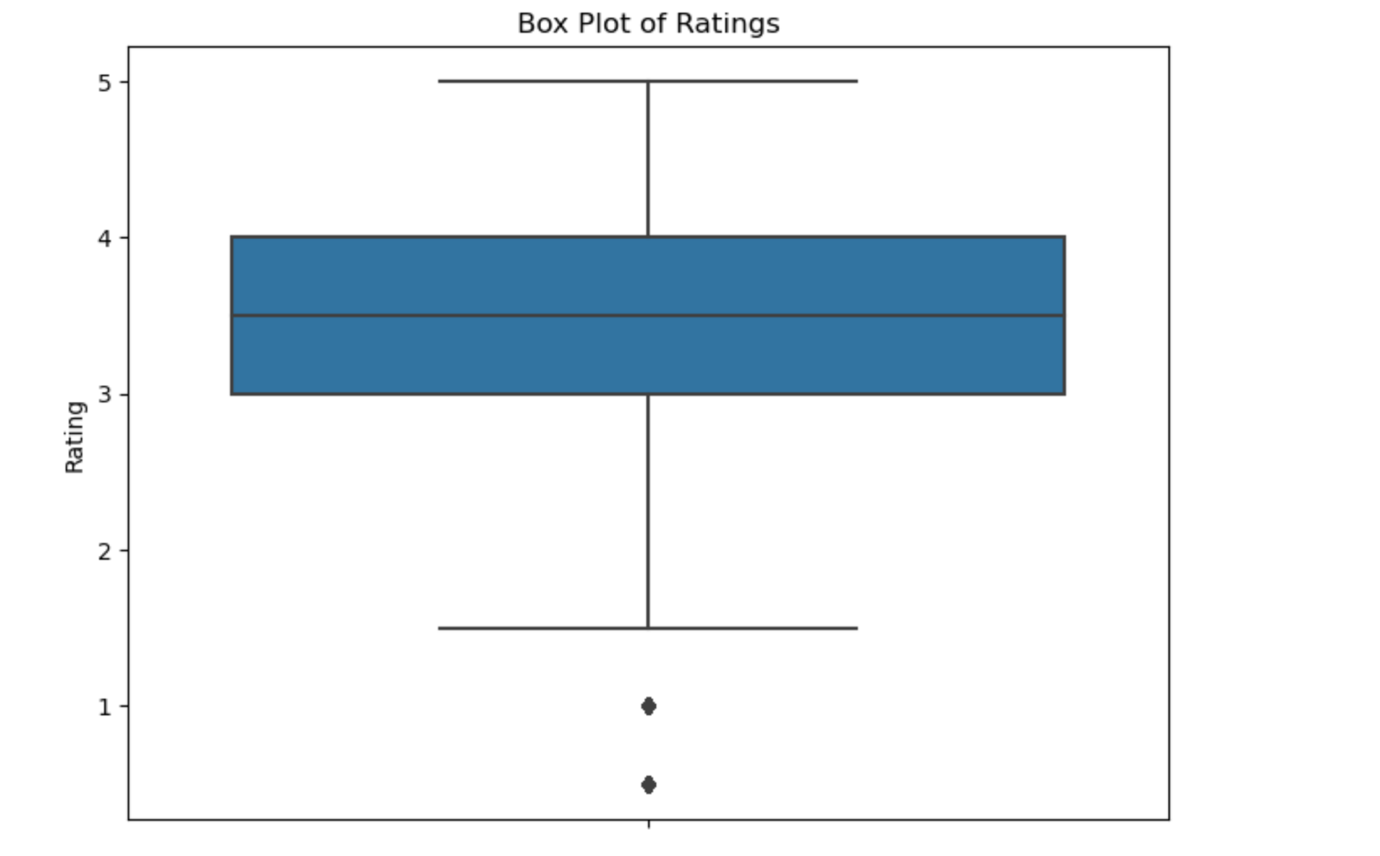
Next we analyzed the “Distribution Of Genres” plot. Drama emerged as the most prevalent genre featuring slightly more than 40,000 movies closely followed by comedy with just under 40,000 films. Action and thriller genres also had significant representation with action being more popular than thriller. Genres like adventure and romance were moderately popular each boasting over 20,000 movies while Sci-Fi and Crime had slightly fewer titles. The less common genres were fantasy, children’s, mystery and horror each with moderate counts but significantly fewer than the top genres. Animation, war, imax musical and western genres had even lower counts with fewer than 10,000 movies each indicating that they cater to more niche audiences. Documentary and film-noir were among the least common genres with fewer than 5,000 movies each. Additionally, there was a small category for movies without any listed genres.

1. Distribution of movies by year



We then looked at the “Distribution Of Movies By Year” which highlighted historical trends in movie production. We observed significant growth in the movie industry starting in the late 1960s peaking in the late 1990s with a maximum output of over 6,000 movies. This was followed by a sharp decline in production suggesting potential shifts in industry dynamics such as changes in movie consumption patterns, technological advancements or economic factors impacting production.

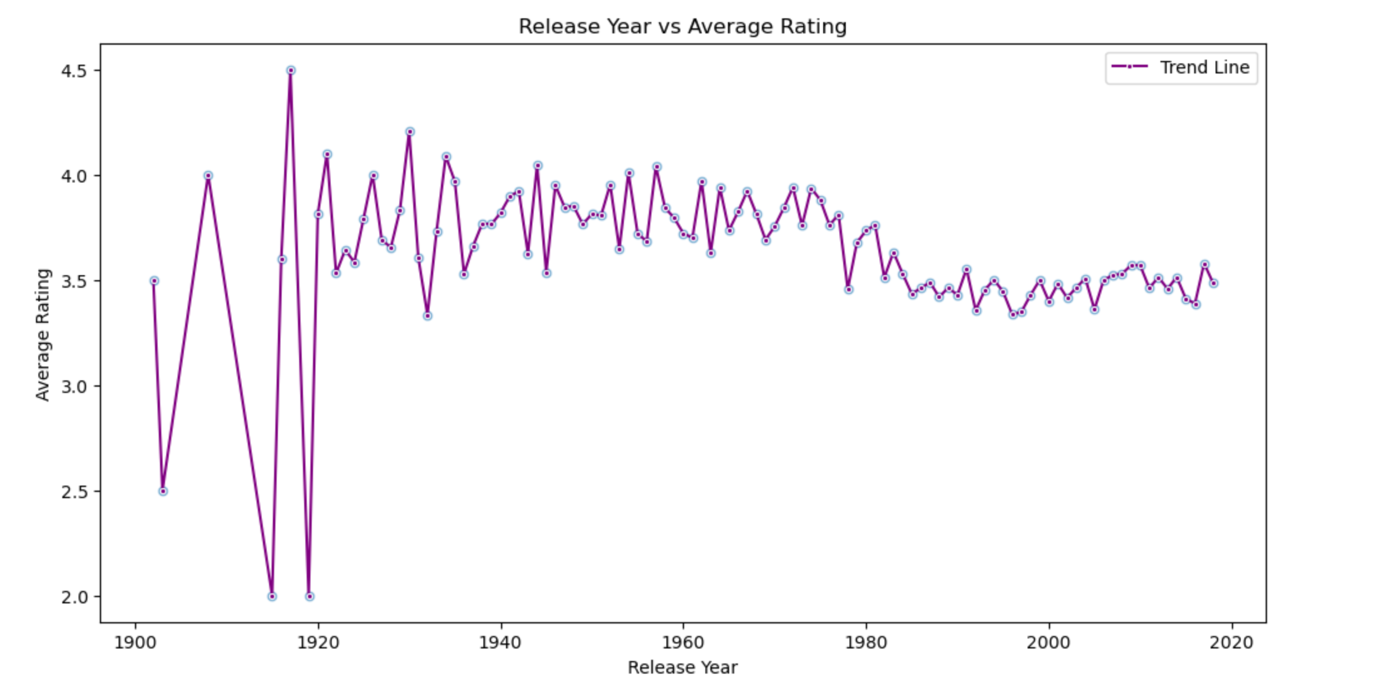
1. Analysis of rating outliers



Finally, the “Analysis Of Rating Outliers” plot identified some unusual ratings. Despite their presence, we decided to keep these outliers in the dataset as they would be valuable in curating our recommender system.

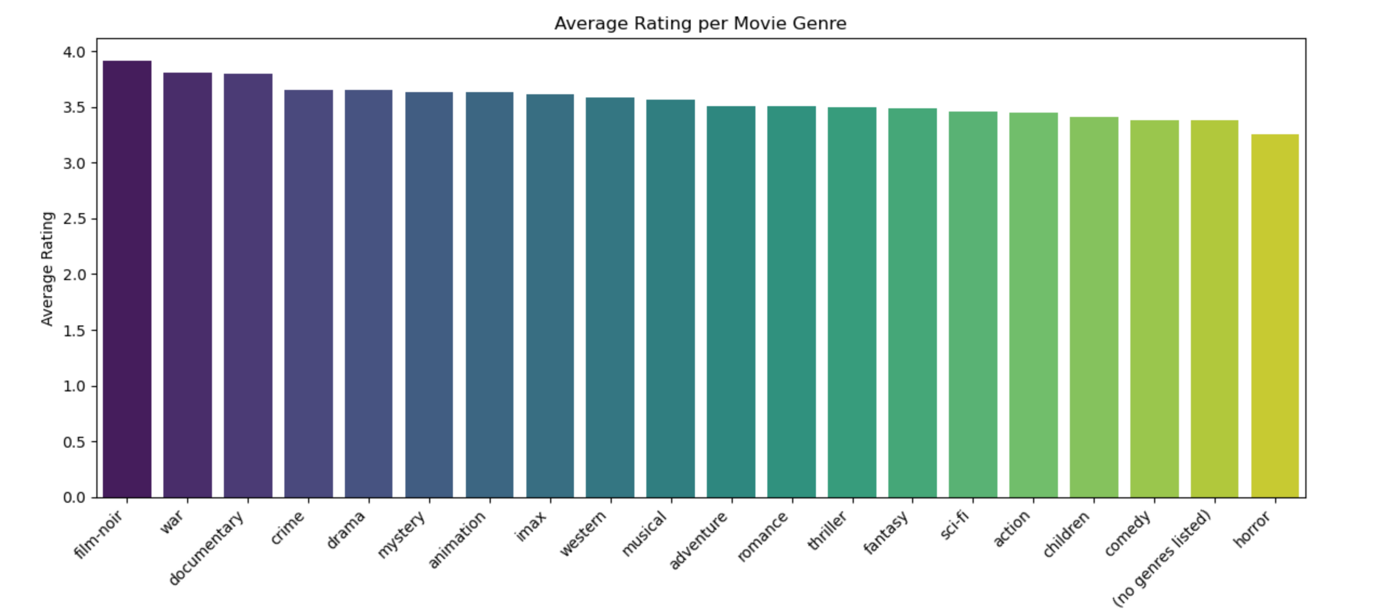
* Bivariate Analysis

1. Release Year vs Average Rating



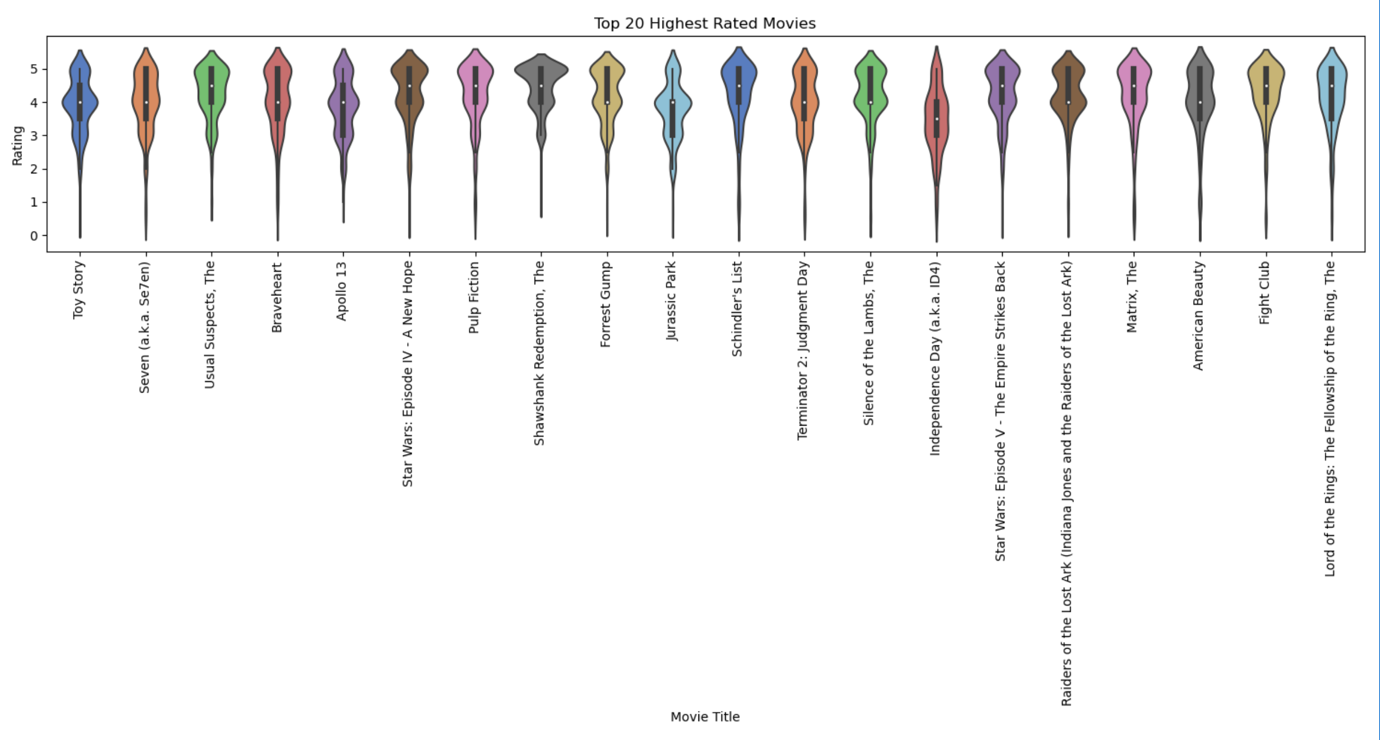
Firstly, we examined the "Release Year Vs Average Rating" plot, which illustrated the fluctuations in average ratings over time. In the early years, from 1900 to 1940, the ratings showed considerable variation, likely reflecting the film industry's formative period. This was followed by notable peaks in the early 1900s, a stabilization phase post-1940, and a gradual decline from the 1970s to 2020. From the 1980s onwards, ratings consistently hovered around 3.5, with outliers becoming less pronounced over time.

1. Average Rating Per Movie Genre



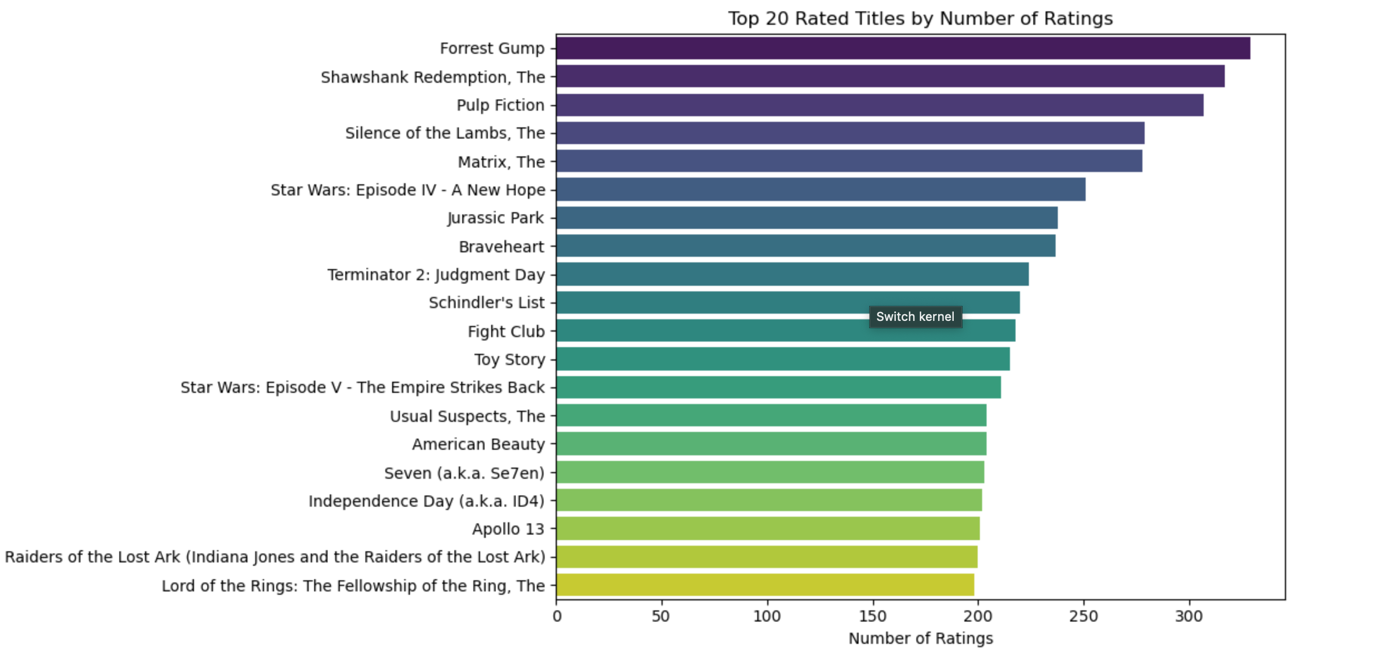
Next, we analyzed the “Average Rating Per Movie Genre” plot which provided insights into how different genres were rated on average. Genres such as film-noir and war received the highest ratings of about 4.0. Documentary, crime and drama genres also scored highly around 3.75. Other genres like mystery, animation, imax, western and musical followed closely. Adventure, romance, thriller and fantasy genres had slightly lower average ratings of around 3.5 while sci-fi, action, children’s and comedy genres averaged closer to 3.0. The genres with the lowest average ratings were unspecified genres and horror movies both falling below 3.0. Overall, film-noir and war emerged as the most highly rated genres while horror films had the lowest average ratings.

1. Top 20 Highest Rated Movies



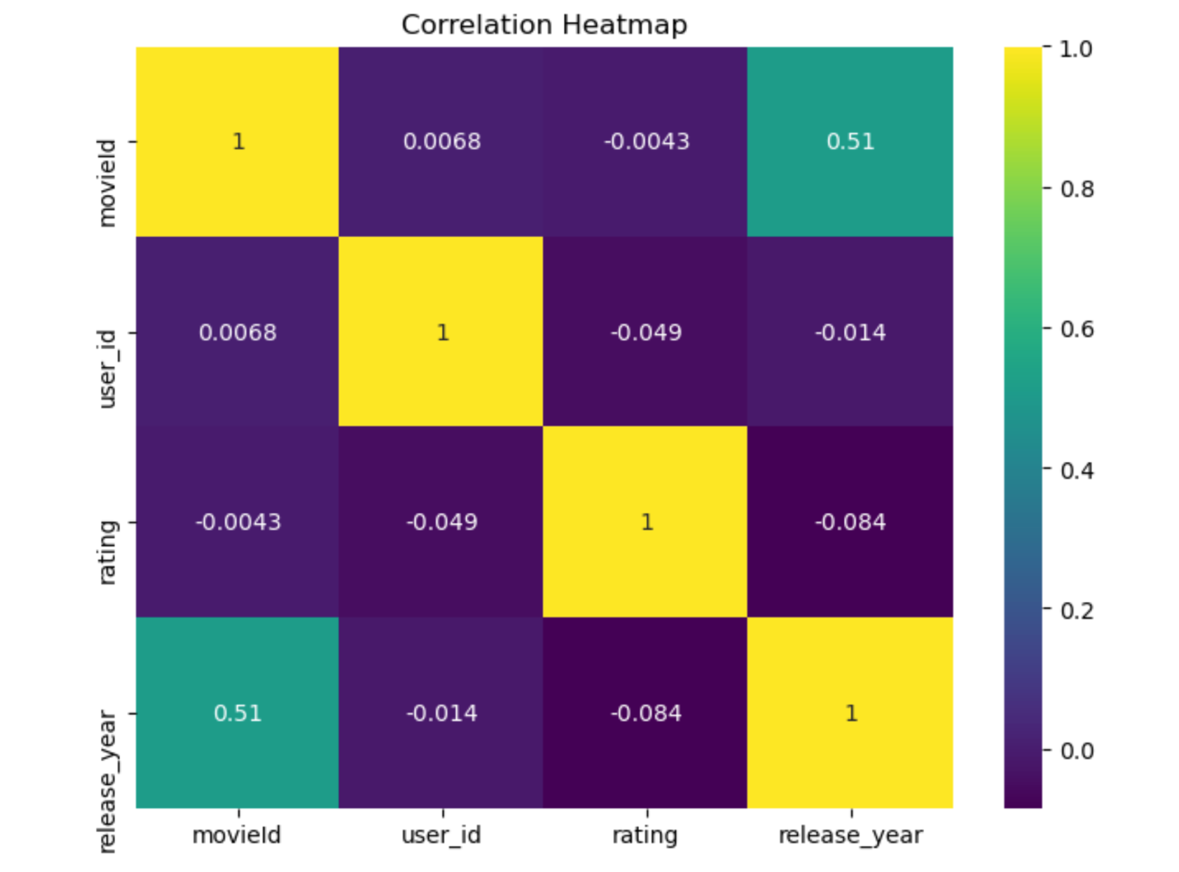
We then reviewed the “Top 20 Highest Rated Movies” plot which illustrated the rating distributions for the top 20 films. Each movie was represented by a vertical violin shape with the width indicating the frequency of ratings at each score level. Most top-rated movies displayed a concentration of high ratings with a bulge near the top of the 5-point scale though there were some variations in rating patterns across different films.

1. Top 20 Rated Titles by Number of Ratings



Lastly, the “Top 20 Rated Titles By Number Of Ratings” plot revealed that 'Forrest Gump' led with the highest number of ratings followed closely by 'The Shawshank Redemption' and 'Pulp Fiction'. The number of ratings ranged from around 200 for 'Lord of the Rings: The Fellowship of the Ring' to nearly 300 for the top-rated films indicating their popularity and wide review base.

1. Correlation Heatmap



To conclude our analysis, we examined the “correlation heatmap” which illustrated the relationships between numeric variables. It revealed a strong positive correlation (0.98) between release year and decade, moderate positive correlations between movieId and both release year (0.51) and decade (0.5) with weak or negligible correlations among other variables. The diagonal values of 1 in the heatmap represented perfect self-correlation.

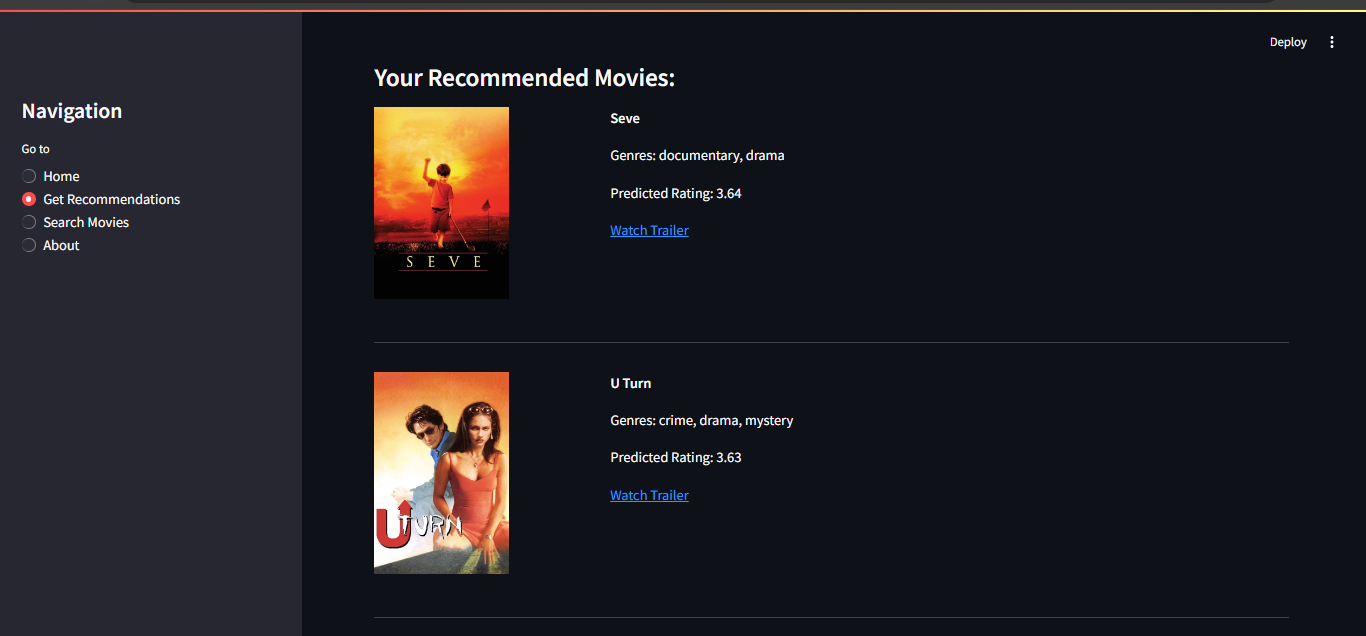
This comprehensive EDA provided valuable insights into trends and relationships within the dataset guiding further analysis and model development.

# MODELING

A dummy model was initially evaluated using the Surprise library as a baseline for the collaborative filtering system generating random predictions based on the observed distribution of ratings. This dummy model was trained and tested resulting in an RMSE of 1.43. GridSearch Cross Validation of the Singular Value Decomposition (SVD) model yielded an RMSE of 0.862 compared to an RMSE of 0.975 for the K-Nearest Neighbors (KNN) model indicating the SVD model's superior performance for collaborative filtering. The SVD model was further assessed through cross-validation achieving a mean RMSE of approximately 0.857 with a very low standard deviation of 0.005 across 5 folds. The `CollabBasedModel` class was utilized to train the SVD model, collect user ratings, generate movie recommendations (including optional genre filtering) and print the top 5 recommendations. TF-IDF vectorization and cosine similarity were integrated within the `ContentBasedModel` class to train a content-based movie recommendation system and provide genre-based recommendations based on similarity to a specified movie title. The `HybridModel` class combined collaborative and content-based filtering using a weighted average to offer hybrid recommendations with user ratings guiding the process.

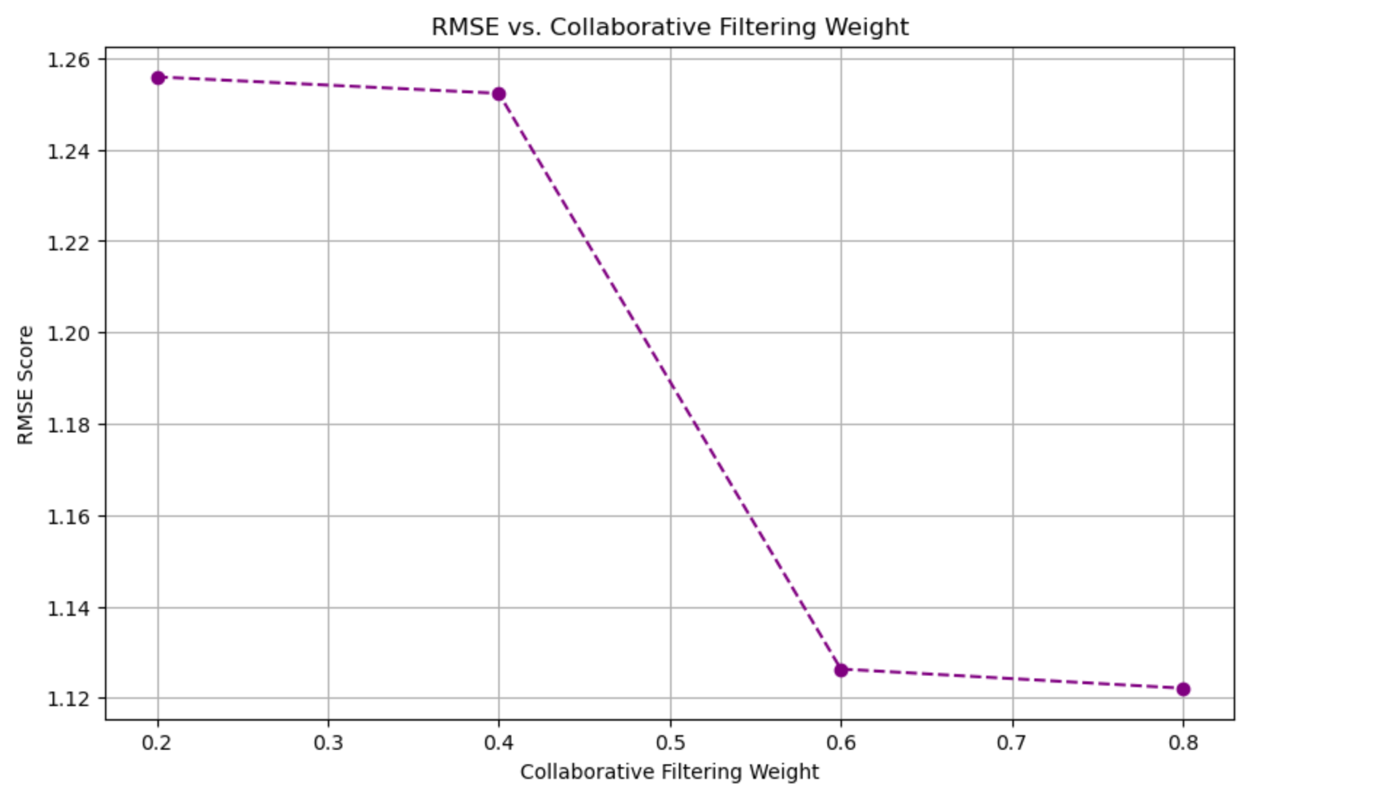
**Deployment**

Streamlit is a Python library used to create web applications for data science projects. In this deployment, it's used to build an interactive movie recommendation system. The app uses a collaborative filtering model based on the Singular Value Decomposition (SVD) algorithm from the Surprise library trained on user-movie ratings. Key features include displaying trending movies, personalized recommendations, movie search functionality and genre-based browsing. The app fetches movie posters from a TMDB API and attempts to show trailers using a YouTube API. It's structured with a sidebar for navigation between home, recommendations, search, and about pages. The model and data are loaded using caching for improved performance. The code demonstrates integration of machine learning models with a user-friendly interface including features like rating movies, receiving personalized recommendations and exploring movies by genre. API keys are handled securely using environment variables. While the app provides a comprehensive movie exploration experience, there's room for optimization as continuous improvement continues on the hybrid model. Below is a snippet from the top recommended movies page:



# CONCLUSIONS

The collaborative filtering model achieved an RMSE score of 0.86 performed well at accurately predicting user preferences based on interaction data. However, the hybrid model, which combined features from both collaborative and content-based methods, showed a higher RMSE suggesting it might not capture user preferences as effectively. The hybrid model benefitted from a greater emphasis on collaborative filtering as indicated by lower RMSE scores with increasing collaborative weights.



This showed that prioritizing collaborative filtering in a hybrid approach enhanced accuracy and recommendation quality more than relying solely on content-based filtering.

# RECOMMENDATIONS

1. Experiment further with collaborative filtering weights in finer increments around the optimal value (e.g., between 0.6 and 0.8) to provide additional insights into achieving even better performance.
2. Implement cross-validation to ensure that the observed improvements in RMSE are consistent across different subsets of the data. This helps in verifying that the results are not due to random chance or overfitting.
3. Enhance the content-based model by incorporating more detailed item features such as plot summaries which could provide value especially for users with limited interaction history.
4. Explore other hyperparameters and configurations for both collaborative filtering and content-based components of the hybrid model to potentially enhance performance.
5. Evaluate the model using additional metrics such as Mean Average Precision (MAP) or Precision@K to gain a more comprehensive understanding of its recommendation quality.
6. Incorporate user feedback and real-world testing to validate the model's effectiveness in practical scenarios and ensure it aligns with user preferences and expectations.
7. Regularly evaluate the recommendation system with updated data and metrics to ensure it adapts to changing user preferences and content.
8. Integrate additional techniques such as deep learning-based model to further enhance the system's capabilities and address any remaining limitations.