MOVIE RECOMMENDER SYSTEM

AUTHORS;

Jeremy Waiguru

Paul Ngatia

Mercy Kiragu

Winfred Kinya

OUTLINE

1. Project Overview
2. Problem Statement
3. Objectives
4. Data Description
5. Analysis
6. Modelling
7. Conclusion
8. Recommendations

PROJECT OVERVIEW

Objective: Develop a versatile movie recommendation system using the following techniques ;

1. **Simple Recommender**: Ranks movies by popularity using metrics like ratings count and user reviews to generate a list of top-rated movies.
2. **Collaborative Filtering**: The main recommender, analyzing user behavior and preferences to suggest movies based on similar users or movies.
3. **Hybrid Recommendation model**: Addresses the cold-start problem by recommending movies based on intrinsic features like genre, director, and actors, useful for new users and items.

Evaluation: Success measured using RMSE and MAE scores to ensure effectiveness and accuracy.

PROBLEM STATEMENT

- ★ BingeMax needs to enhance user satisfaction and engagement by developing a robust movie recommendation system. This system will suggest the top 5 movies to each user based on their ratings of other films, utilizing the MovieLens "small" dataset.
- ★ The effectiveness of this system is crucial for improving user interaction, retention, and overall engagement on the platform.

KEY STAKEHOLDERS

Users:

- Receive personalized movie recommendations, improving satisfaction and engagement.

BingeMax Management:

- Seeks to enhance user experience and maintain a competitive edge in the entertainment market.

Data Scientists and Developers:

- Responsible for developing and implementing the recommendation algorithm.

Content Providers and Studios:

- Benefit from increased visibility and targeted recommendations of their movies to interested users.

OBJECTIVES

- ★ **Build a recommender system with above an 80% accuracy.**
- ★ **Evaluate Algorithm Efficiency**: Ensure the recommendation system trains quickly, uses resources efficiently, and delivers real-time recommendations without excessive computational demands.
- ★ **Address Cold Start Problem**: Assess the system's ability to provide relevant suggestions for new users with limited or no historical data.
- ★ **Analyze Impact on User Retention**: Compare retention rates before and after implementing the recommender system to determine its effect on user continuation.
- ★ **Evaluate User Engagement**: Monitor changes in user interaction, such as time spent on the platform, click-through rates, and movies watched per session.
- ★ **Increase User Engagement and Reduce Churn**: Deploy a personalized recommendation system aiming to boost user engagement by 25% and reduce churn rate by 15%.

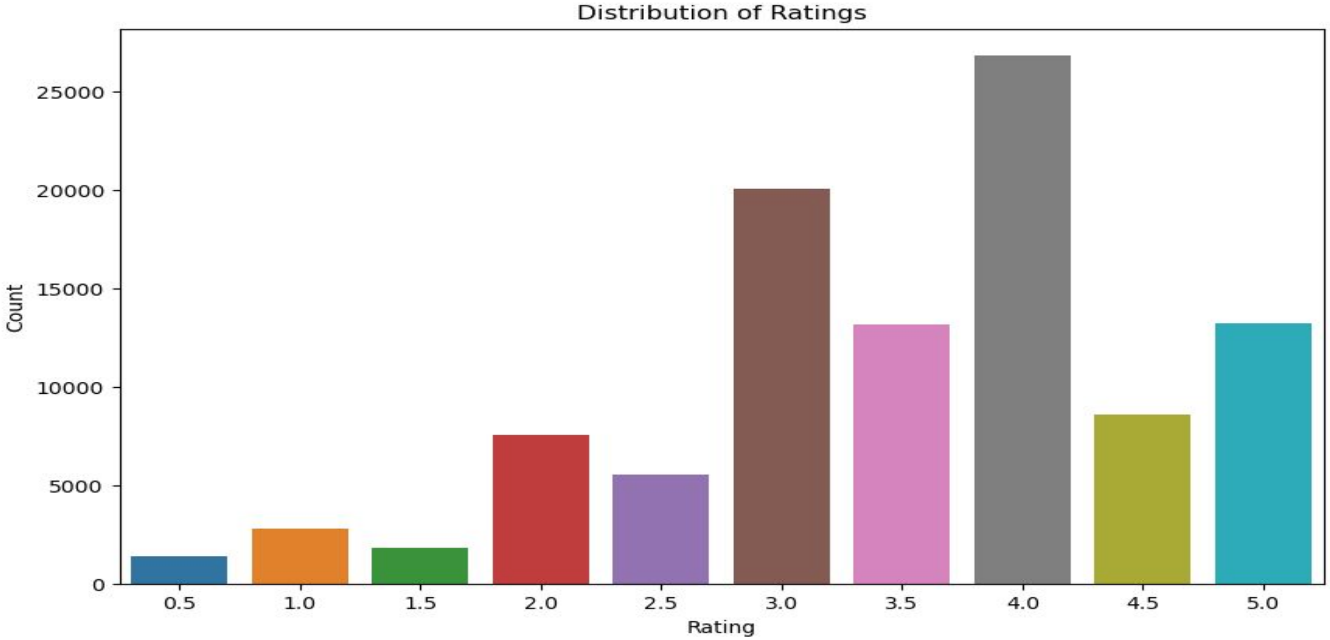
DATA DESCRIPTION

Dataset: MovieLens
Files: movies.csv, ratings.csv, links.csv, tags.csv
Key Variables:

- **userId**: Identifier for the user
- **movieId**: Identifier for the movie
- **rating**: User rating (e.g., 1 to 5 stars)
- **timestamp**: When the rating was given (Unix epoch format)
- **title**: Movie title
- **genres**: List of genres for the movie

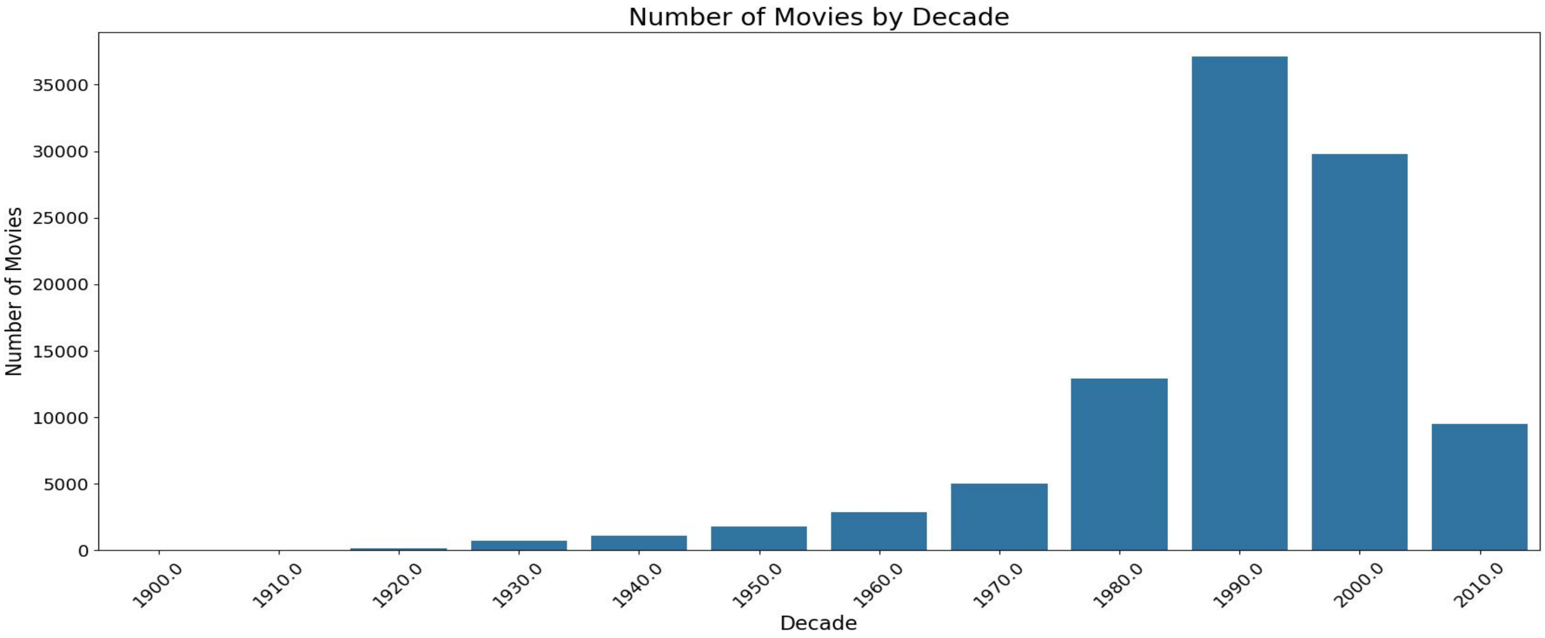
Source: MovieLens Dataset

ANALYSIS
1. Distribution of Movie ratings.



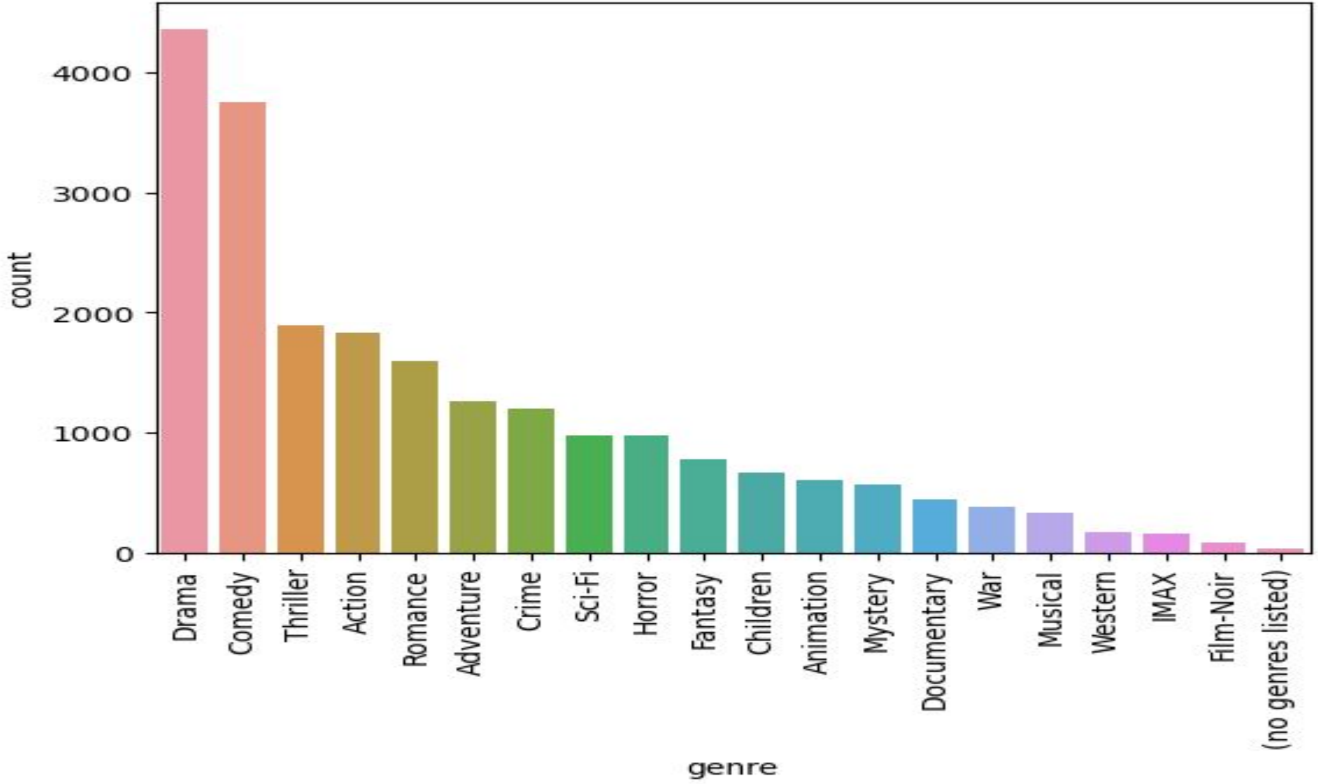
The top-rated movies received a score of 5, while the lowest-rated ones were rated as low as 0.5. Most films were rated either 3 or 4, with a minority receiving ratings of 0.5 or 1.5.

2. Analysis of movies production by decade.



The production of movies gradually rose from the 1920s, reaching its peak in the 1990s, before declining in the early 2000s to levels comparable to those seen in the 1980s.

3. Distribution of movie genres



MODELLING

- **Popularity Model:**
  - **RMSE:** 0.90
  - Acts as a baseline by recommending items based on overall popularity or user rating counts.
- **Collaborative Filtering Models:**
  - **SVD**
    -**RMSE:** 0.8153
  - **SVD (Singular Value Decomposition) After Regularization:**
    - **RMSE:** 0.8104
    - SVD with regularization (n_factors = 20, reg_all = 0.05) performed exceptionally well, yielding the lowest RMSE.
    - Regularization slightly reduced overfitting and improved generalization, enhancing model performance.
  - **KNNBaseline:**
    - **RMSE:** 0.8185
    - Utilizes baseline estimates and similarity metrics for recommendations.
- **KNNWithMeans:**
  - **RMSE:** 0.8382
  - Considers item mean ratings, resulting in a higher RMSE compared to SVD and KNNBaseline.
- **KNNBasic:**
  - **RMSE:** 0.9083
  - Basic collaborative filtering with higher RMSE, indicating less effective recommendations compared to other models.

Best model;

Hybrid Recommender:

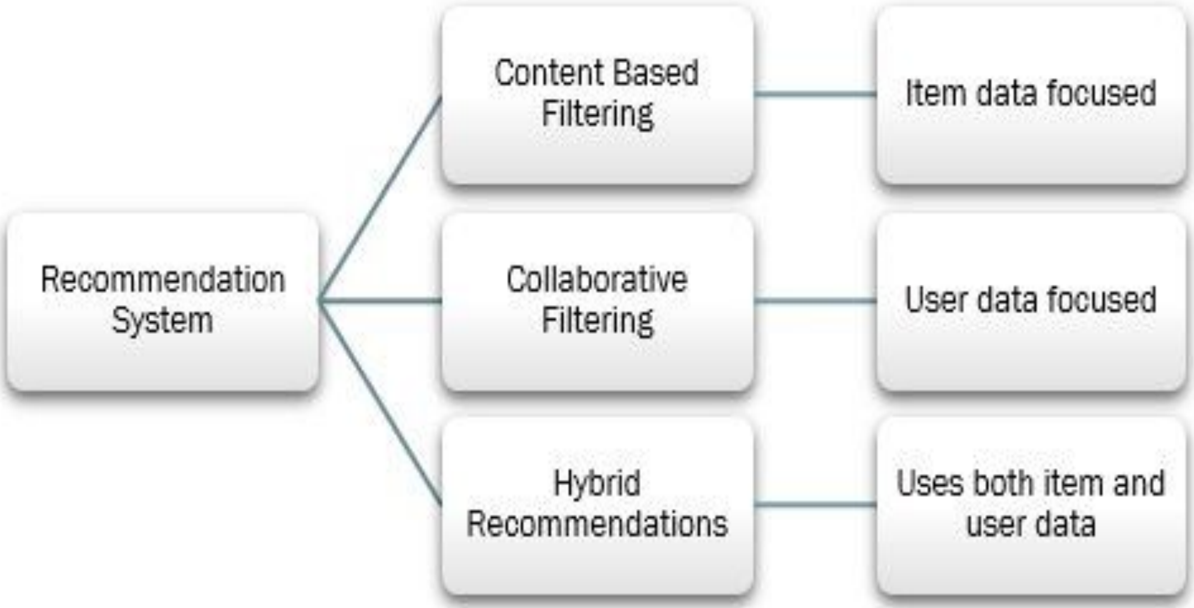
- **RMSE:** 0.2886
- Focuses on item features and user profiles for recommendations.
- While not as accurate as collaborative filtering models, it is particularly useful for addressing the cold-start problem by recommending items based on content similarity rather than user interactions

CONCLUSION

- ❖ The Hybrid recommendation model proved to be the most accurate with an RMSE of 0.2886 which is the lowest rmse.
- ❖ The hybrid recommendation system, which combines collaborative filtering (SVD) with content-based filtering, effectively leverages the strengths of both methods.
- ❖ This integrated approach enhances recommendation accuracy, handles the cold-start problem, and provides a more personalized user experience.

RECOMMENDATIONS

1. **Enhance User Engagement:**
   - Bingemax should Incorporate user feedback mechanisms to refine recommendations..
2. **Leverage Marketing Insights:**
   - Use recommendation data to target marketing and promotions effectively.
3. **Ensure Data Privacy:**
   - Implement robust privacy measures and communicate policies clearly.
4. **Partner with Content Creators:**
   - Collaborate with studios for early access to new releases.
5. **Build Community:**
   - Foster user interaction through reviews, lists, and forums.
6. **Monitor Performance:**
   - Track system performance in real-time and address issues promptly.
7. **Plan for Scalability:**
   - Design the system to handle growth and expanding data efficiently.



Any Questions ??


THANK YOU!