Intelligent Systems

Movie Recommendation System

GROUP 7

- ALEX MWERA
- NOEL SEDA
- ZENA LISA KARARI

Introduction Overview

This project addresses the need for effective movie recommendations by comparing two **distinct approaches**: content-based and collaborative filtering methodologies.

Project Objectives

IDENTIFY KEY FACTORS

To understand the essential components influencing recommendation accuracy.

ENHANCE USER EXPERIENCE

Aim to improve personalization for movie recommendations based on user preferences.

EVALUATE METHODOLOGIES

Assess the effectiveness of content-based and collaborative filtering approaches.

Dataset Overview

The **MovieLens 100k** dataset contains 100,000 ratings from 943 users on 1,682 movies, allowing for extensive analysis of recommendation algorithms.

Content-Based Filtering

Content-based filtering recommends movies by analyzing items' features and user preferences, focusing on the **similarity of item characteristics** to enhance user experience.

Collaborative Filtering Approaches

Collaborative filtering utilizes user preferences to recommend movies, leveraging techniques such as user-based and item-based methods through various libraries for implementation.

Hybrid Approach

COMBINING TECHNIQUES

Utilizing both content-based and collaborative filtering enhances recommendation accuracy.

IMPROVED USER EXPERIENCE

A hybrid model tailors recommendations to individual user preferences effectively.

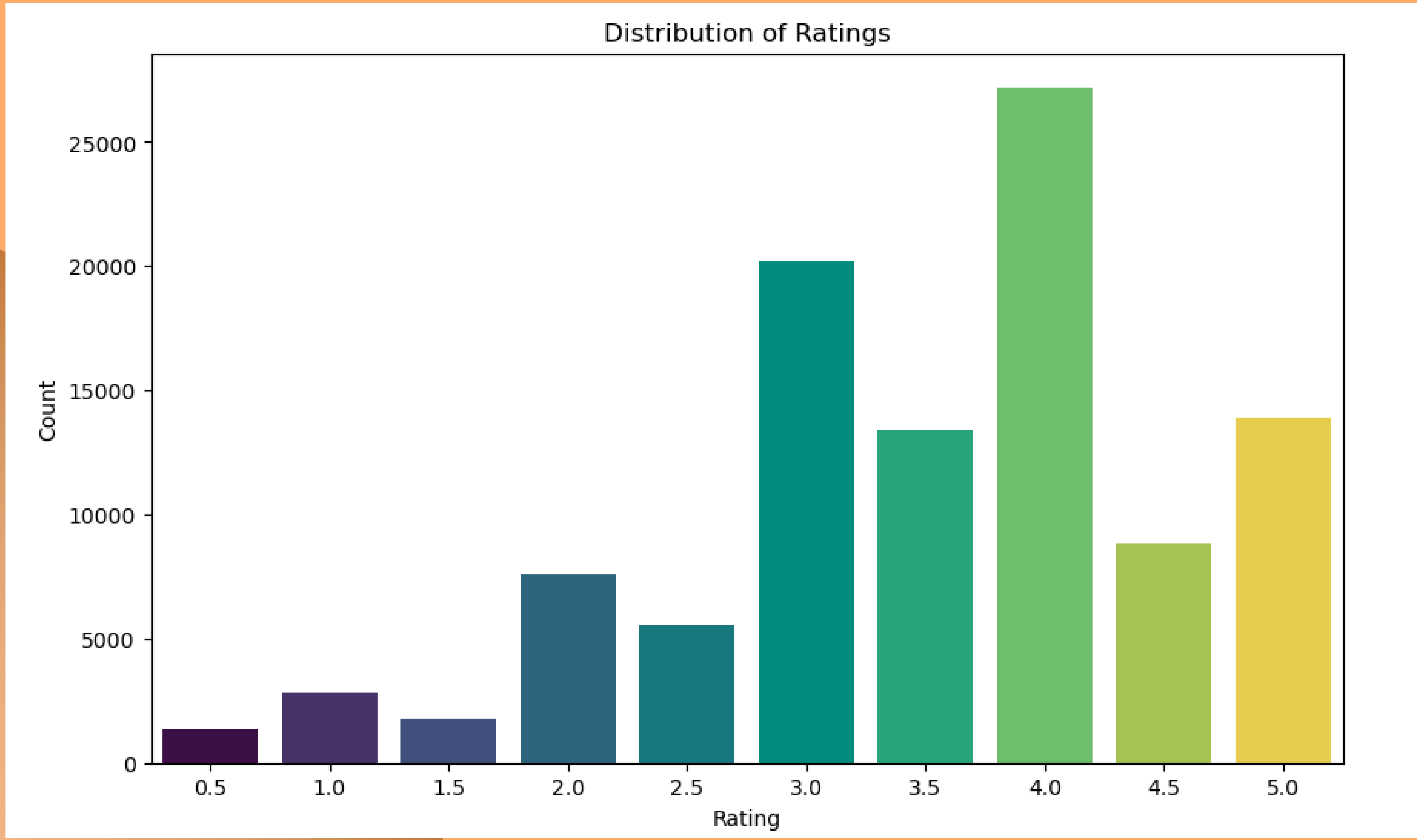
INCREASED SCALABILITY

The approach efficiently handles large datasets while maintaining performance.

EXPLORATORY DATA INSIGHTS

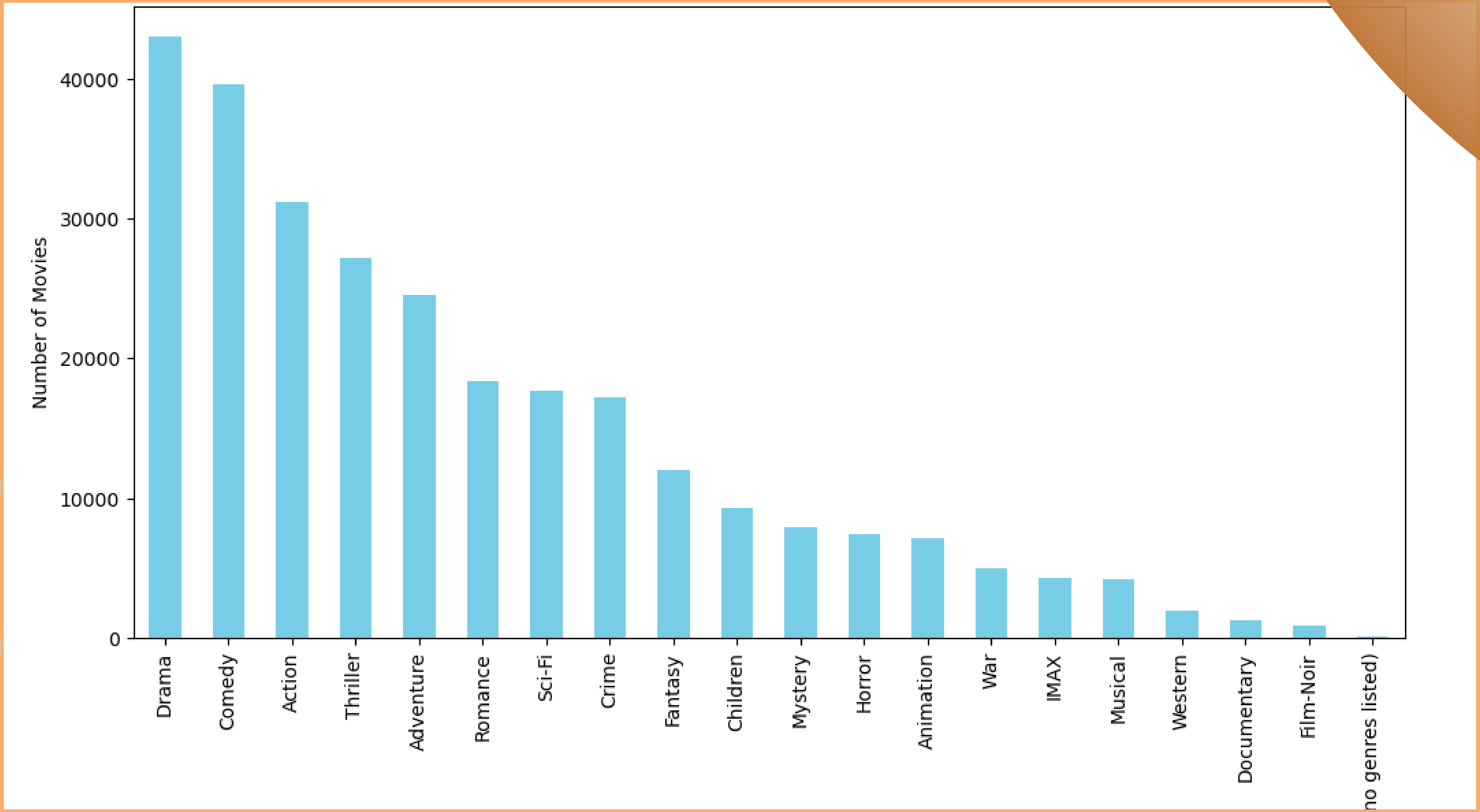
DISTRIBUTION OF RATINGS:

- Most ratings fall within the 3.0 to 5.0 range, indicating generally positive user feedback.
- Highest rating frequency observed at 4.0, suggesting a moderate satisfaction trend.



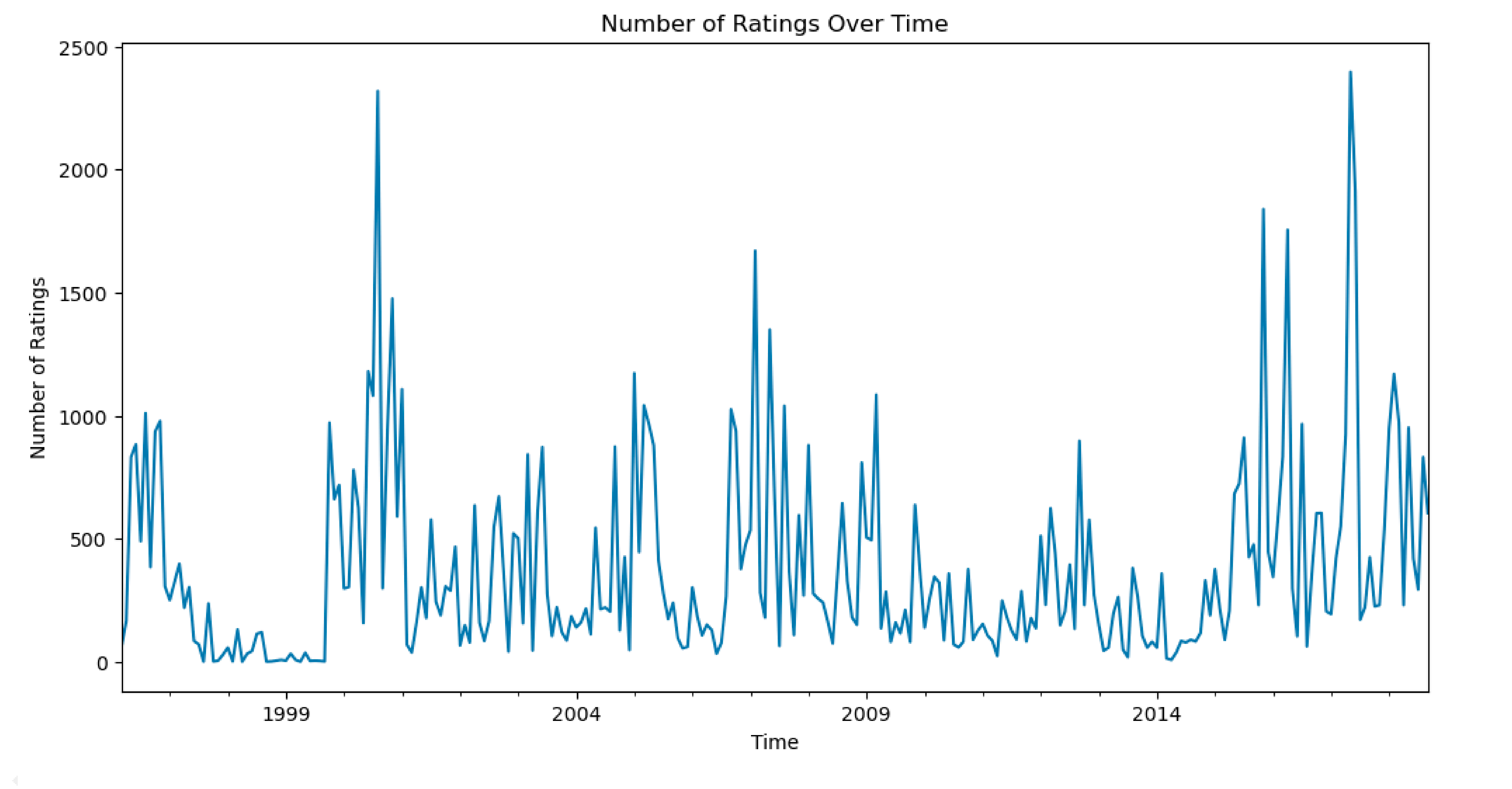
GENRE DISTRIBUTION:

- Drama, Comedy, and Action are the most represented genres in the dataset.
- Suggests a content diversity that allows wide-ranging personalization.



RATINGS OVER TIME:

- Shows fluctuations in user activity over years, with peaks likely corresponding to popular movie releases.
- Highlights temporal dynamics important for time-aware recommendation.



CONTENT-BASED FILTERING OUTPUT:

- Recommends movies with similar content (Using genre).
- Top results for Toy Story (1995) are mostly animated, family-friendly films.
- Reflects the system's ability to match by features rather than user ratings.

```
def get_recommendations(title, cosine_sim=cosine_sim):
    idx = movies_full_data[movies_full_data['title'] == title].index[0]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:11]  # Get top 10 similar movies
    movie_indices = [i[0] for i in sim_scores]
    return movies_full_data.iloc[movie_indices][['title', 'genres', 'rating']].reset_index(drop=True)
```

```
print(get_recommendations('Toy Story (1995)'))
```

```
                                               title  \
0                                   Toy Story 2 (1999)
1                                Monsters, Inc. (2001)
2                                          Antz (1998)
3        Adventures of Rocky and Bullwinkle, The (2000)
4                       Emperor's New Groove, The (2000)
5                             Shrek the Third (2007)
6                           The Good Dinosaur (2015)
7  Asterix and the Vikings (Astérix et les Viking...
8                       Tale of Despereaux, The (2008)
9                                         Moana (2016)
```

ITEM-TO-ITEM COLLABORATIVE FILTERING OUTPUT:

- Finds movies that are similar to those rated by the user, based on shared rating patterns.
- Top recommendations reflect how similar movies are rated by the community:
  - Toy Story 2 (1999), Groundhog Day (1993), Independence Day (1996).
  - Personalised to the user's input through weighted similarity scores.

```
# Collects user-rated movies interactively, uses get_similar_movies() to generate scores and displays
recommend_from_user_input()
```

```
Enter the movies you've watched and your rating (0 to 5). Type 'done' when finished.


Top Recommendations for You:

1. Toy Story 2 (1999) (Score: 1.15)
2. Groundhog Day (1993) (Score: 0.90)
3. Independence Day (a.k.a. ID4) (1996) (Score: 0.90)
4. Willy Wonka & the Chocolate Factory (1971) (Score: 0.89)
5. Mission: Impossible (1996) (Score: 0.88)
6. Nutty Professor, The (1996) (Score: 0.88)
7. Bug's Life, A (1998) (Score: 0.86)
8. Lion King, The (1994) (Score: 0.86)
9. Babe (1995) (Score: 0.85)
10. Monsters, Inc. (2001) (Score: 0.83)
```

Key Challenges in Recommendation Systems

Cold Start

Cold start occurs when there is **insufficient data** for new users or items.

Data Sparsity

Data sparsity limits the **availability of user-item interactions**, affecting recommendations.

Scalability

Scalability addresses the **system's ability** to handle increasing data volume and complexity.

Conclusion & Future Wor

In conclusion, our project highlights the effectiveness of both filtering methods, suggesting further exploration into **hybrid approaches** for improved recommendations.