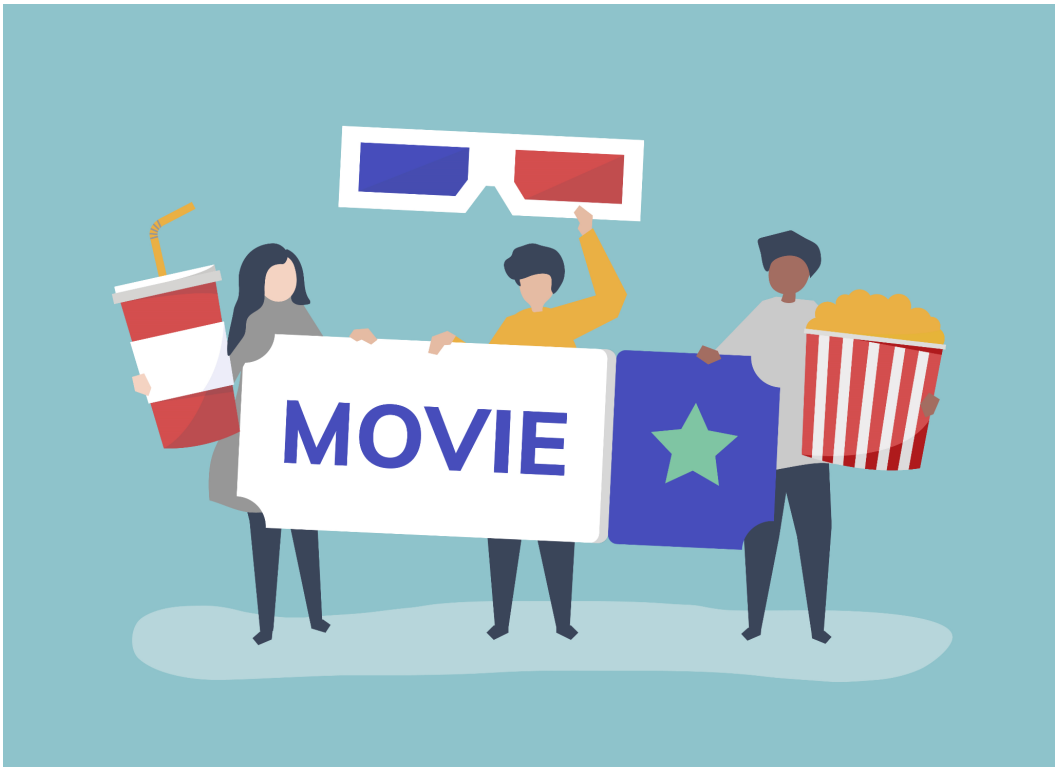
MOVIE RECOMMENDATION SYSTEM

IT Workshop Project



SUBMITTED BY:

Ayushi Dubey (01601192022)

Azmeen Khatoon (01701192022)

Bhumika Gupta (01801192022)

SUBMITTED TO:

Mr. Santanoo

TABLE OF CONTENTS

ABSTRACT

The movie recommendation system plays a crucial role in enhancing user experiences and driving user engagement in the era of extensive content consumption. This project aimed to develop a movie recommendation system using the Item-Based Collaborative Filtering (IBCF) approach. The system leverages user ratings data to identify similarities between movies and generate personalized recommendations for users.

INTRODUCTION

The era of digital entertainment has witnessed an exponential growth in the availability of movies and a vast array of content. However, with such an overwhelming selection, users often struggle to discover movies that align with their personal preferences. This challenge has led to the development of movie recommendation systems, which leverage advanced algorithms to provide tailored suggestions, enhancing the movie-watching experience and facilitating content exploration. In this project, we aim to build a movie recommendation system using item-based collaborative filtering techniques implemented in R.
The primary objective of our project is to create a recommendation engine that suggests relevant movies to users based on their viewing history and preferences. By analyzing the patterns of movie ratings and user-item interactions, our system will identify similarities between movies and users to generate accurate recommendations. Through this, we strive to address the issue of information overload and enable users to discover movies that resonate with their tastes and interests.

The motivation behind this project stems from the growing demand for personalized experiences in the entertainment industry. A robust movie recommendation system has the potential to revolutionize the way users explore and engage with movies, leading to increased customer satisfaction and user retention. Additionally, such systems have the capacity to benefit e-commerce platforms by driving sales, enhancing user engagement, and fostering customer loyalty.

To achieve our objectives, we will utilize the recommenderlab package in R, which provides a comprehensive set of tools for building recommendation systems. By employing item-based collaborative filtering techniques, we will leverage user-item ratings to calculate similarities and generate personalized movie recommendations.

Furthermore, we will evaluate the performance of our system using metrics such as accuracy and precision to assess the effectiveness of our recommendations.

In the subsequent sections of this report, we will delve into the data pre-processing steps, implementation of the recommendation model, evaluation of the system's performance, and the insights gained from our analysis. Through this project, we aim to contribute to the field of movie recommendation systems and provide a practical example of building an effective recommendation engine using R.

DATA DESCRIPTION

The dataset used for building our movie recommendation system is the MovieLens Dataset. This dataset comprises of 105339 ratings applied to 10329 movies. It contains information about user ratings, movie IDs, movie titles, genres, and timestamps.

The movie_data dataset provides details about the movies in the dataset. It includes the movieId, which serves as a unique identifier for each movie, the movie title, and the genres associated with each movie. The summary of this dataset shows that the movieId ranges from 1 to 149532, with a mean value of 31924. The dataset consists of 10329 movies, with titles and genres stored as character variables.

The ratings_data dataset contains information about the ratings given by users to different movies. It includes the userId, movieId, rating, and timestamp. The summary of this dataset reveals that the userId ranges from 1 to 668, with a mean value of 364.9. The movieId ranges from 1 to 149532, with a mean value of 13381. The ratings range from 0.5 to 5, with a mean rating of 3.517. The timestamp indicates the time at which the rating was recorded.

These datasets serve as the foundation for our movie recommendation system. By analyzing the ratings and movie information, we aim to generate personalized movie recommendations for users based on their preferences and viewing history.

METHODOLOGY

1. DATA ACQUISITION

The MovieLens Dataset was obtained and used as the primary source of data for this project. The dataset consists of 105339 ratings applied to 10329 movies, providing a

comprehensive pool of user preferences and movie information.

2. DATA PREPARATION

This is conducted in three steps:

1. Selecting useful data
2. Normalizing data
3. Binarizing the data

Data Selection:

Through this we visualized the top users and movies through a heatmap. Then we visualized the distribution of the average ratings per user.

Data Normalization:

In the case of some users, there can be high ratings or low ratings provided to all of the watched films. This will act as a bias while implementing the model. In order to remove this, we normalized the data. Normalization is a data preparation procedure to standardize the numerical values in a column to a common scale value. This is done in such a way that there is no distortion in the range of values. Normalization transforms the average value of our ratings column to 0. We then plotted a heatmap that portrays our normalized ratings.

Data Binarization:

In the final step of the data preparation, in this data science project, we binarized the data. Binarizing the data means that we have two discrete values 1 and 0, which will allow the recommendation system to work more efficiently. We defined a matrix that will consist of 1 if the rating is above 3 and otherwise it will be 0.

3. DATA PREPROCESSING

- The movie_data and ratings_data were loaded into the R environment.
- The data was explored to gain insights into its structure and summary statistics.
- Data cleaning techniques were applied to handle missing values or inconsistencies, ensuring the dataset's integrity.

4. DATA TRANSFORMATION

- To incorporate movie genres into the recommendation system, a one-hot encoding technique was employed.
- The movie_data was split into multiple columns, with each column representing a genre category.
- A binary value of 1 was assigned to a movie if it belonged to that particular genre, and 0 otherwise.

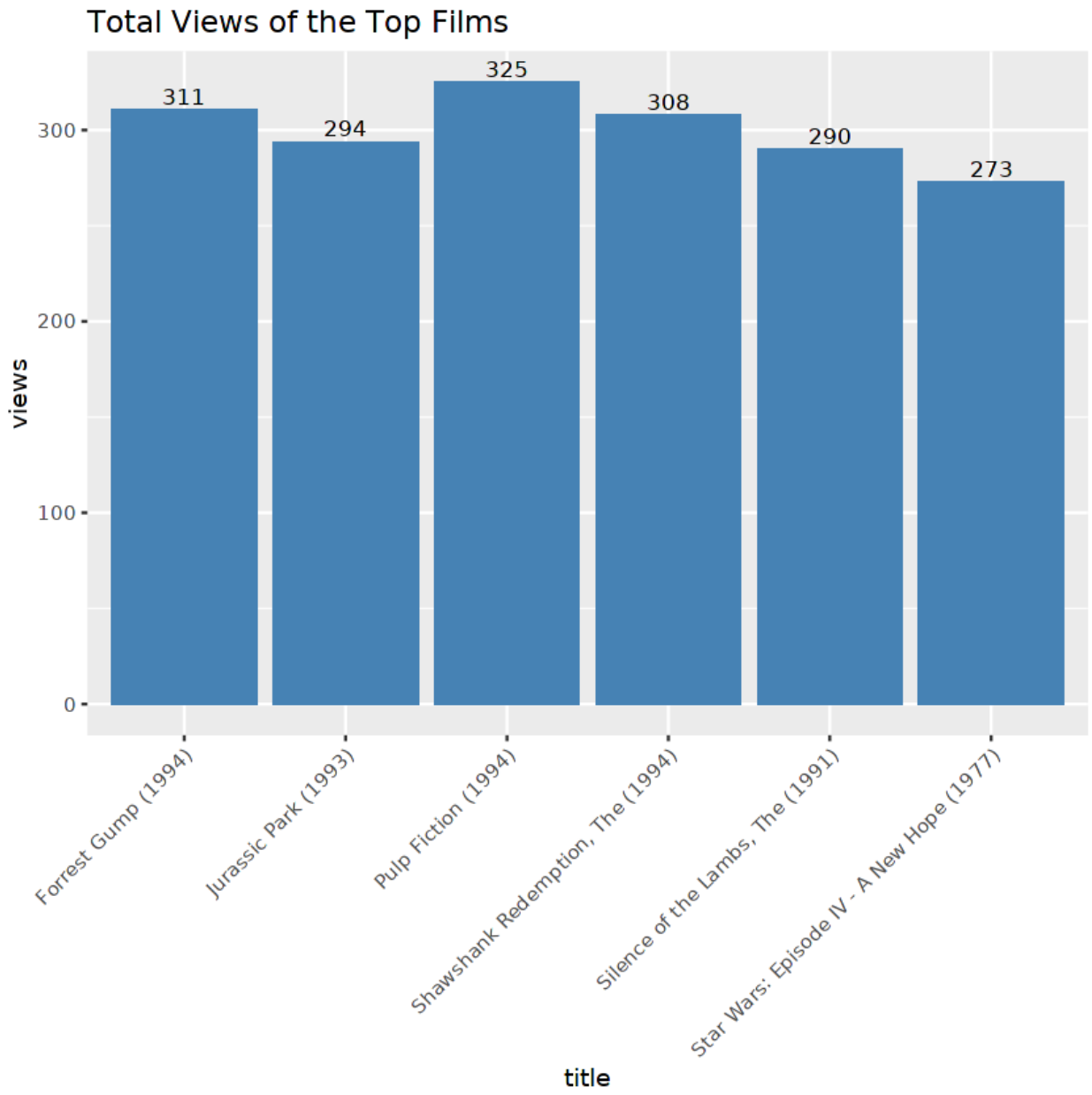
5. BUILDING THE RECOMMENDATION SYSTEM

- The ratings_data was transformed into a rating matrix, with users as rows and movies as columns.
- The rating matrix was converted into a sparse matrix, which is a suitable format for collaborative filtering.
- Collaborative filtering, specifically the Item-Based Collaborative Filtering (IBCF) method, was applied.
- Similarity calculations were performed between movies to identify similar items.
- Recommendations were generated based on the similarity between movies and user preferences.

DATA ANALYSIS AND VISUALIZATION

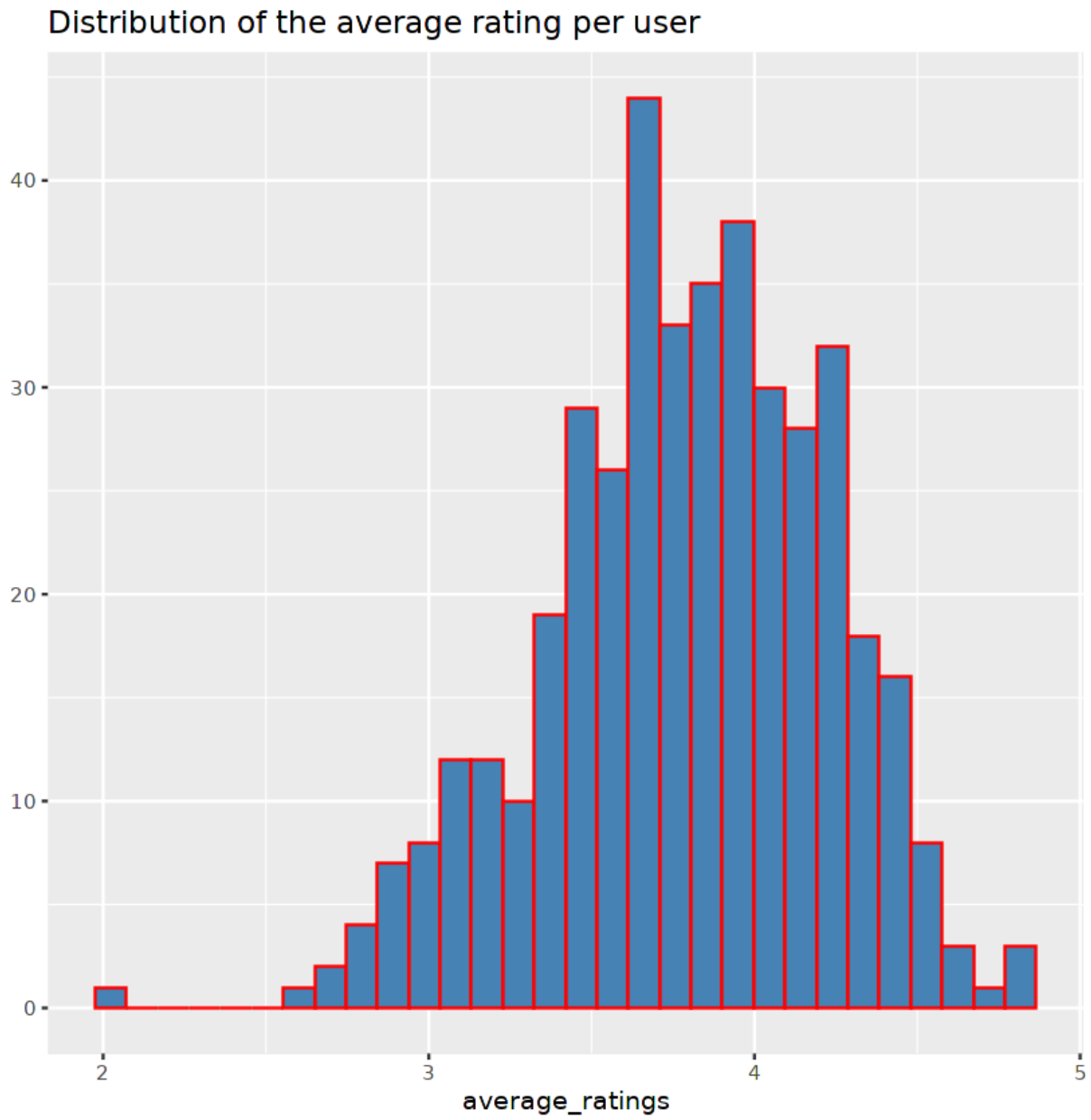
1. Bar Plot for Total Views of Top Films

A bar plot was generated that visualizes the total views for the top films, with movie titles displayed on the x-axis and view counts represented by the heights of the bars.



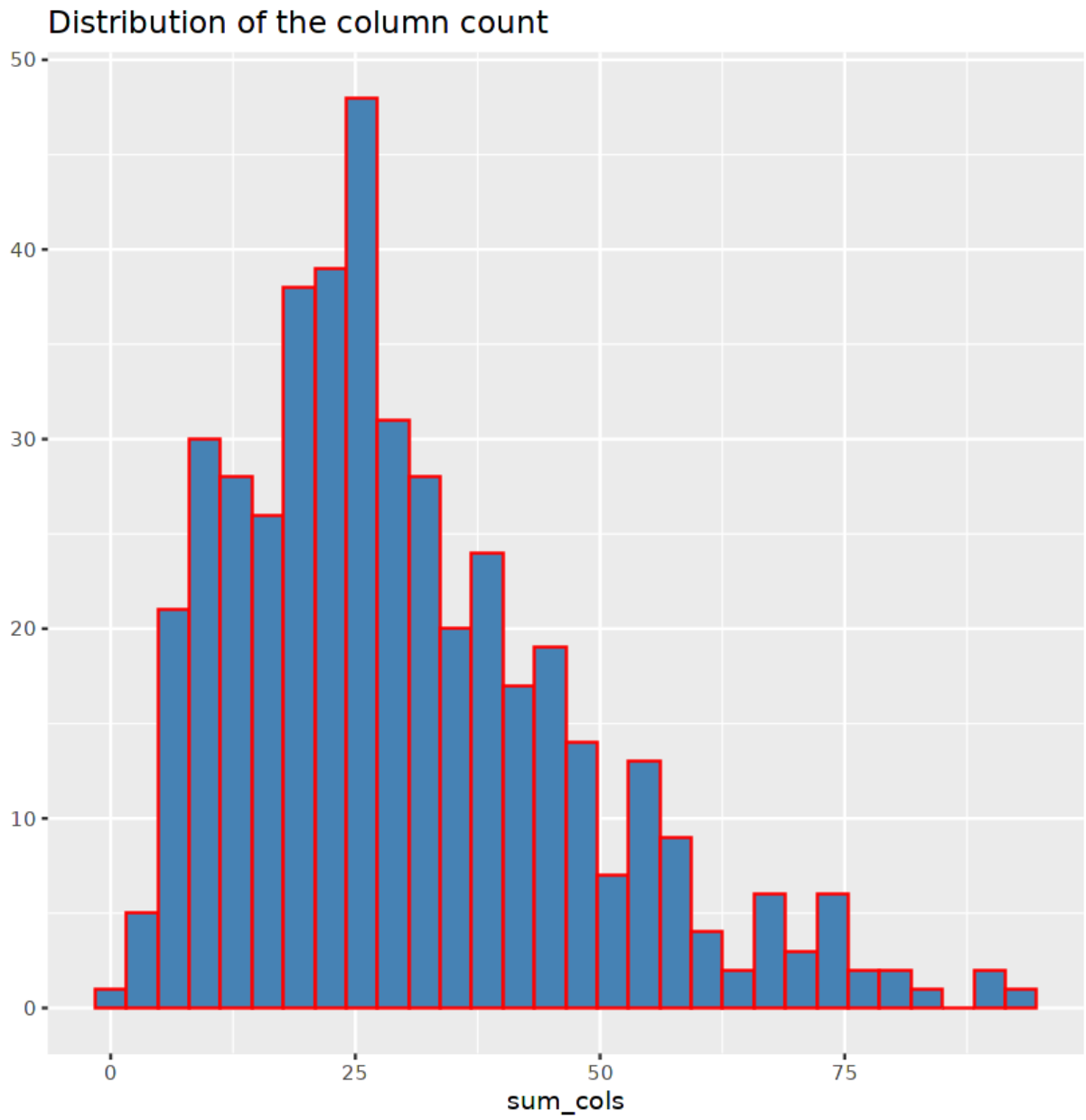
2. Distribution of the Average Rating Per User

A plot showing the distribution of the average rating per user was generated.



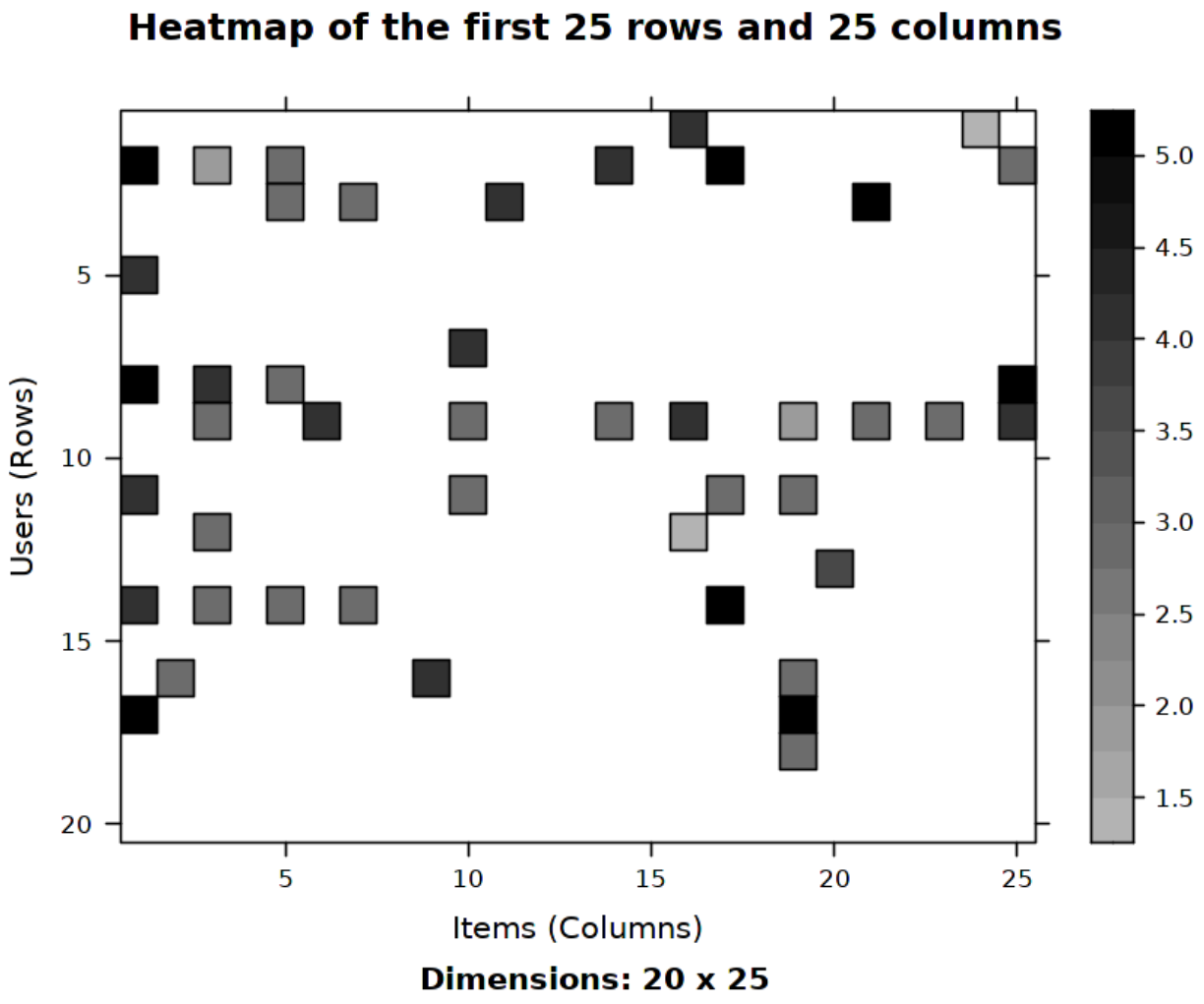
3. Distribution of the Column Count

A histogram using the qplot() function was created to visualize the distribution of the column count.



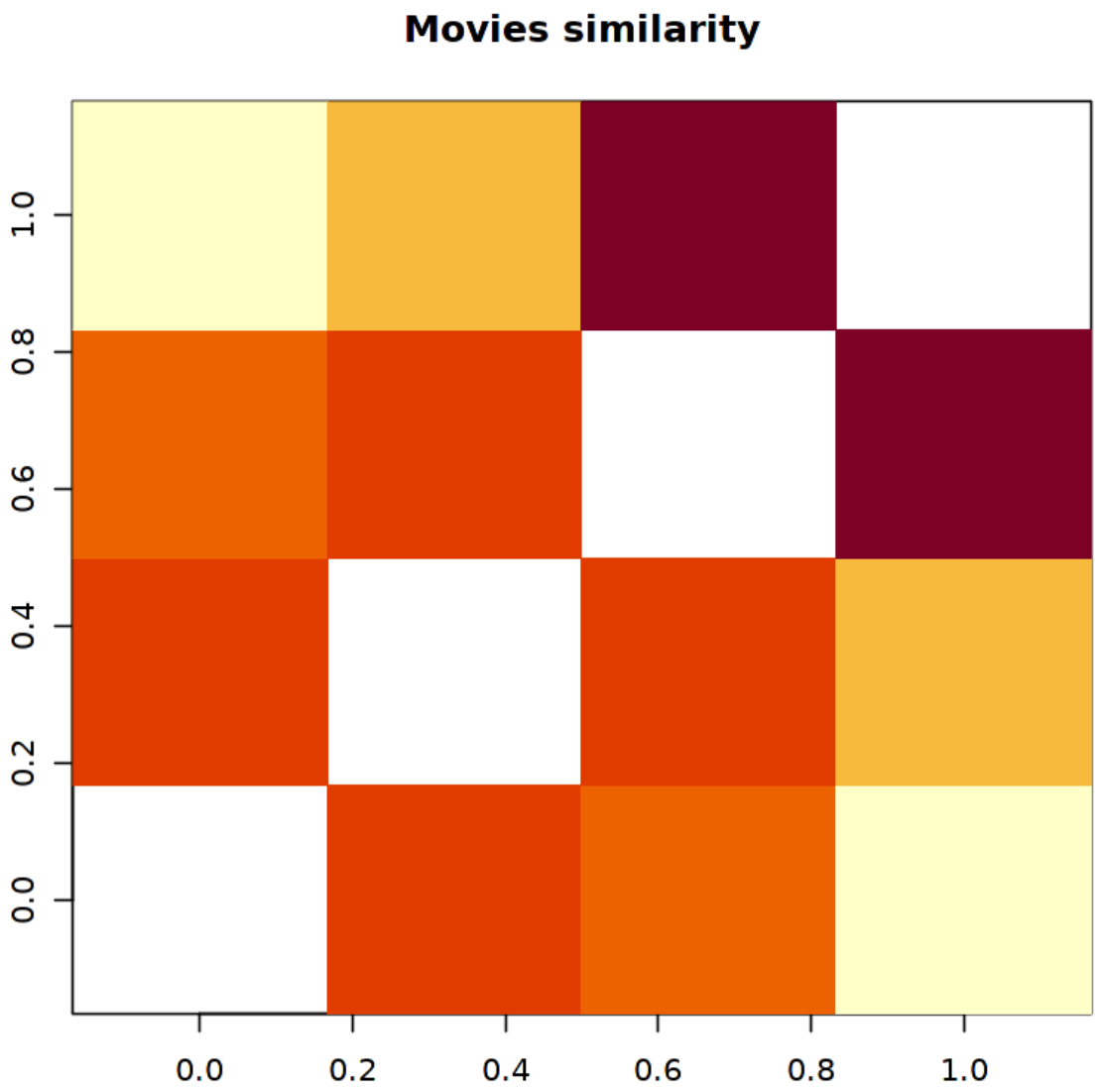
4. Heatmap of Movie Ratings

We obtained the heatmap of the specified subset of the ratingMatrix.



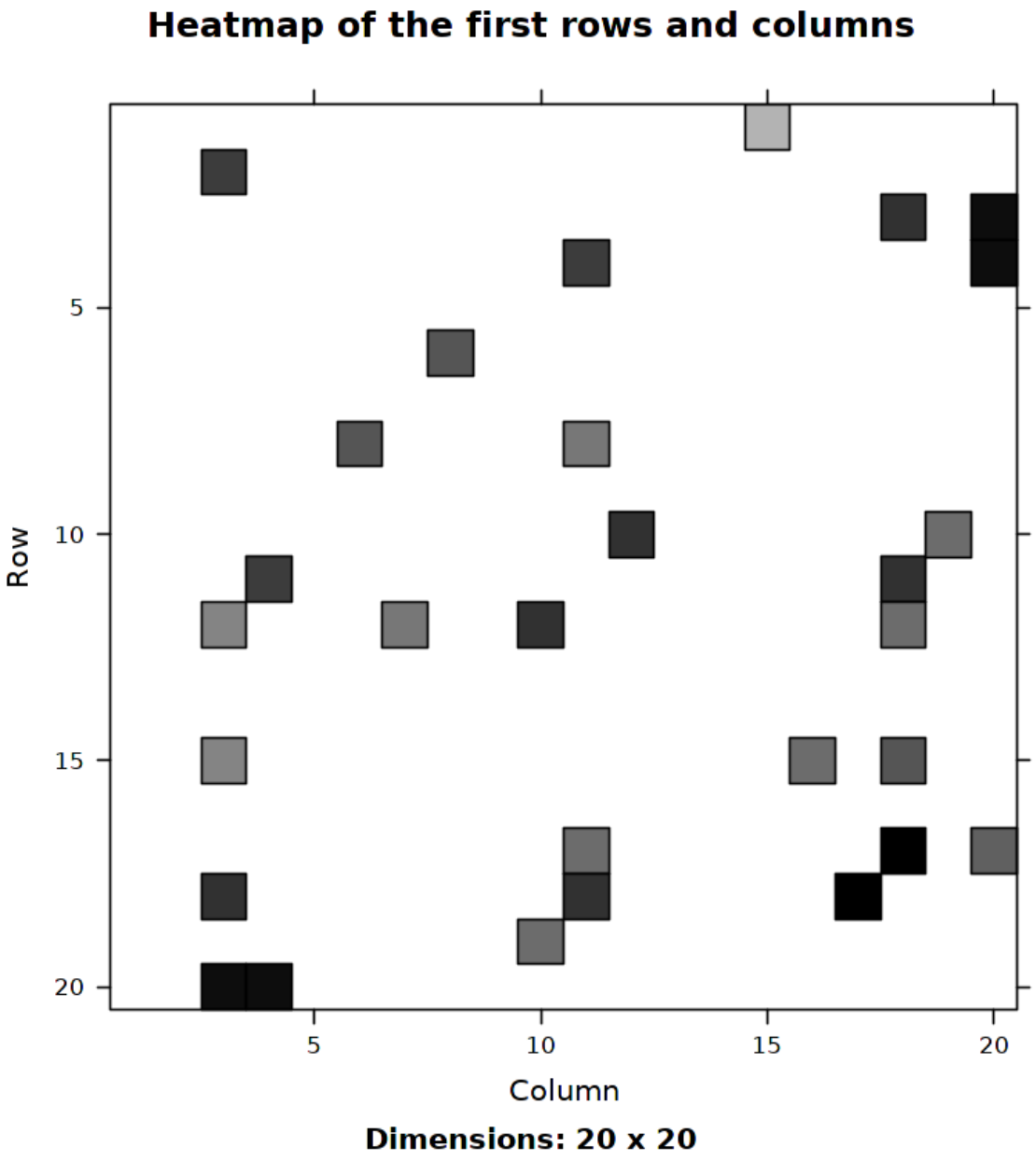
5. Heatmap of Movies Similarity

A heatmap of the movie similarities was displayed, where darker colors indicate higher similarity values between movies.



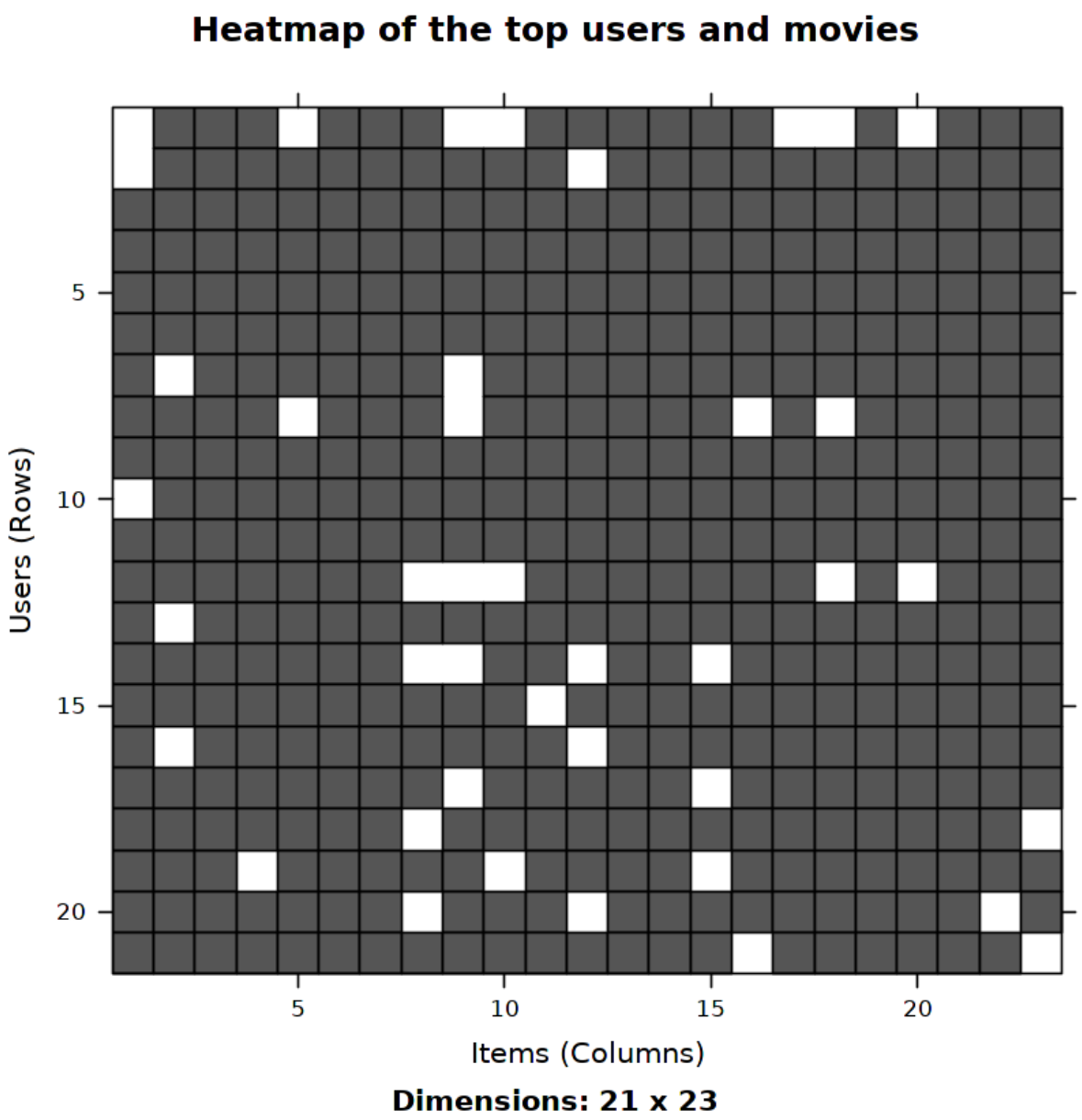
6. Heatmap of the First Rows and Columns

A heatmap plot that visualizes the similarity matrix of the trained recommendation model was generated.



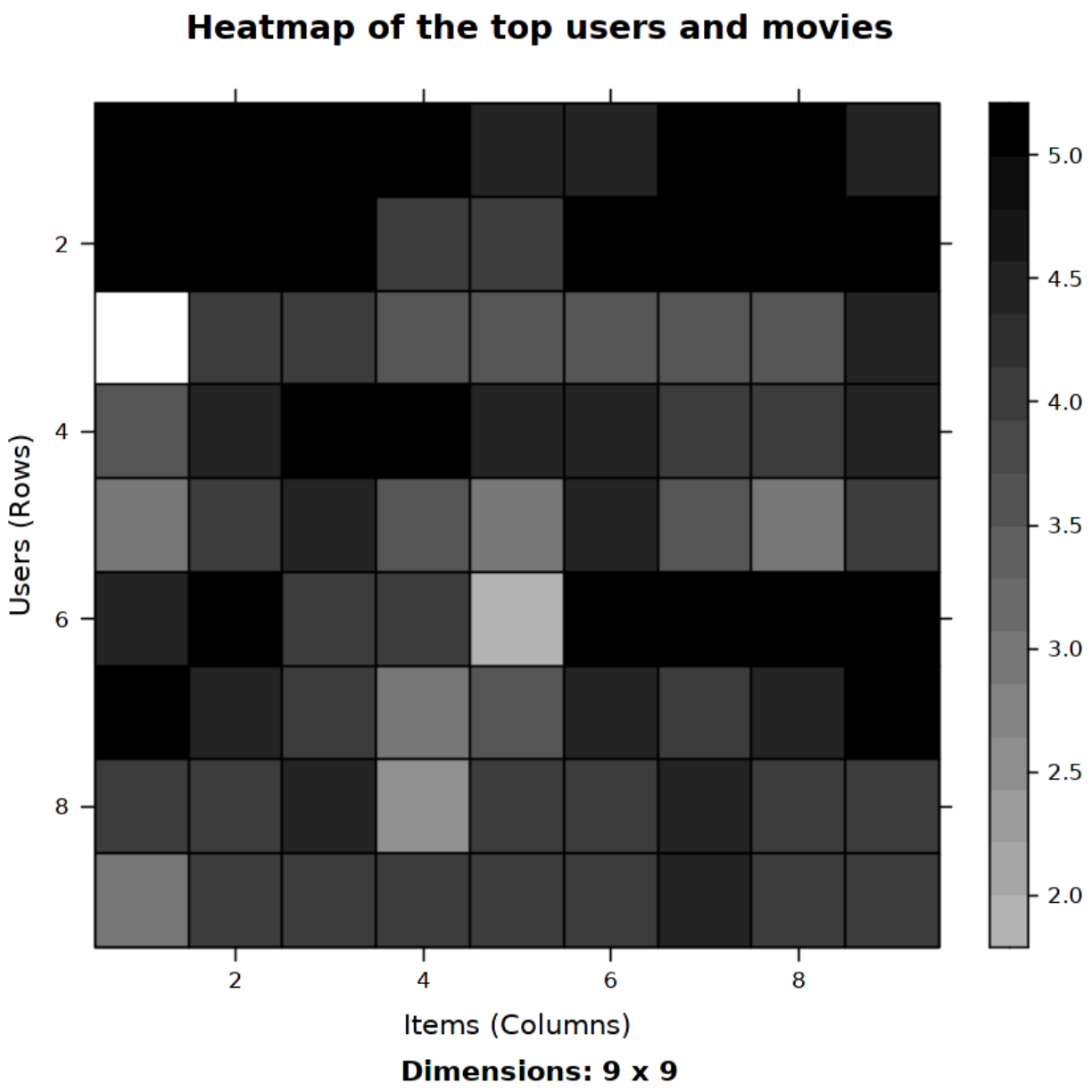
7. Heatmap of the Top Users and Movies After Binarization

A heatmap visualization of the top users and movies based on the binary ratings in the good_rated_films matrix was generated.



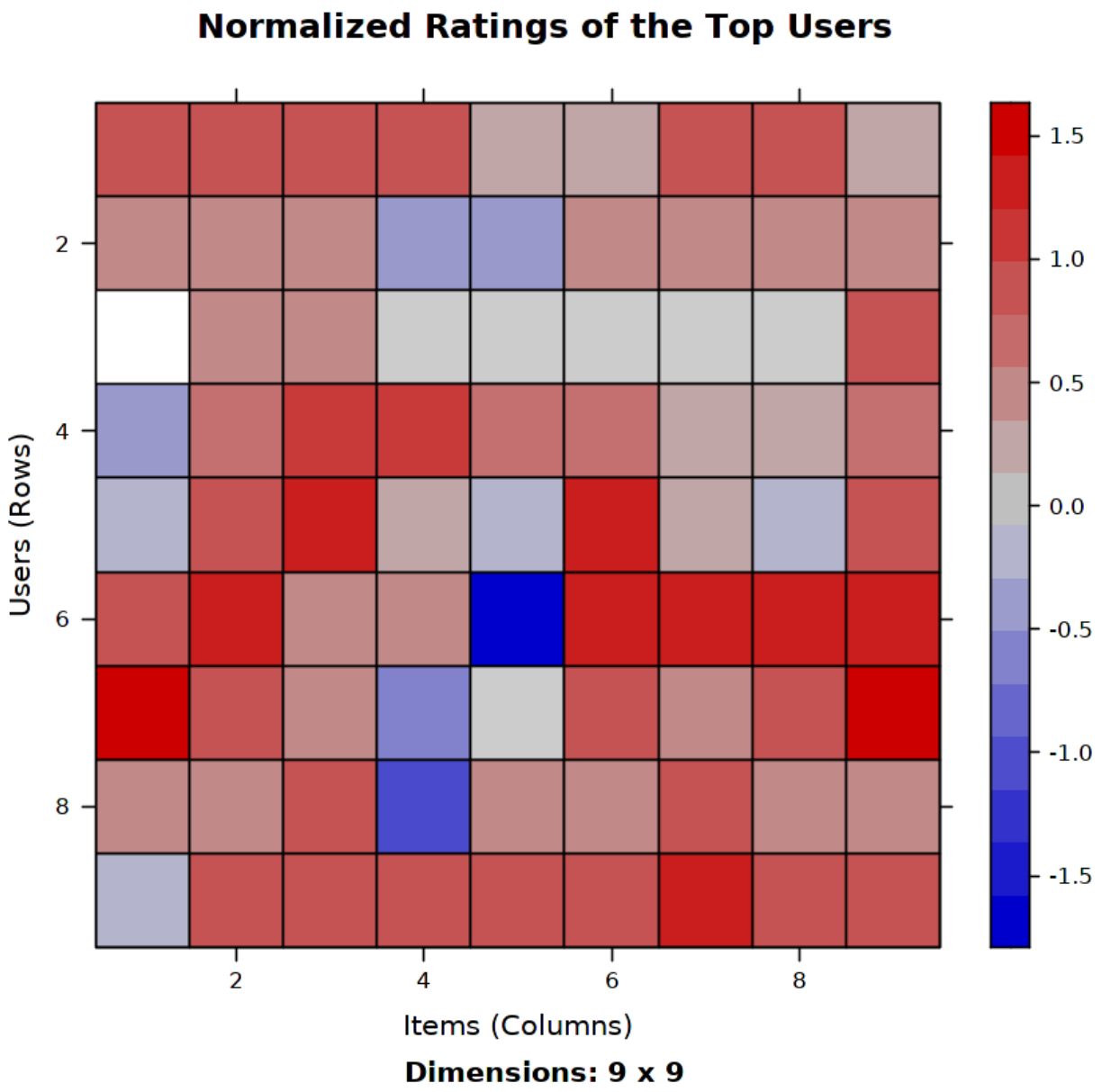
8. Heatmap of Top Users and Movies

A heatmap of top users and movies was generated depicting the movieRatings matrix.



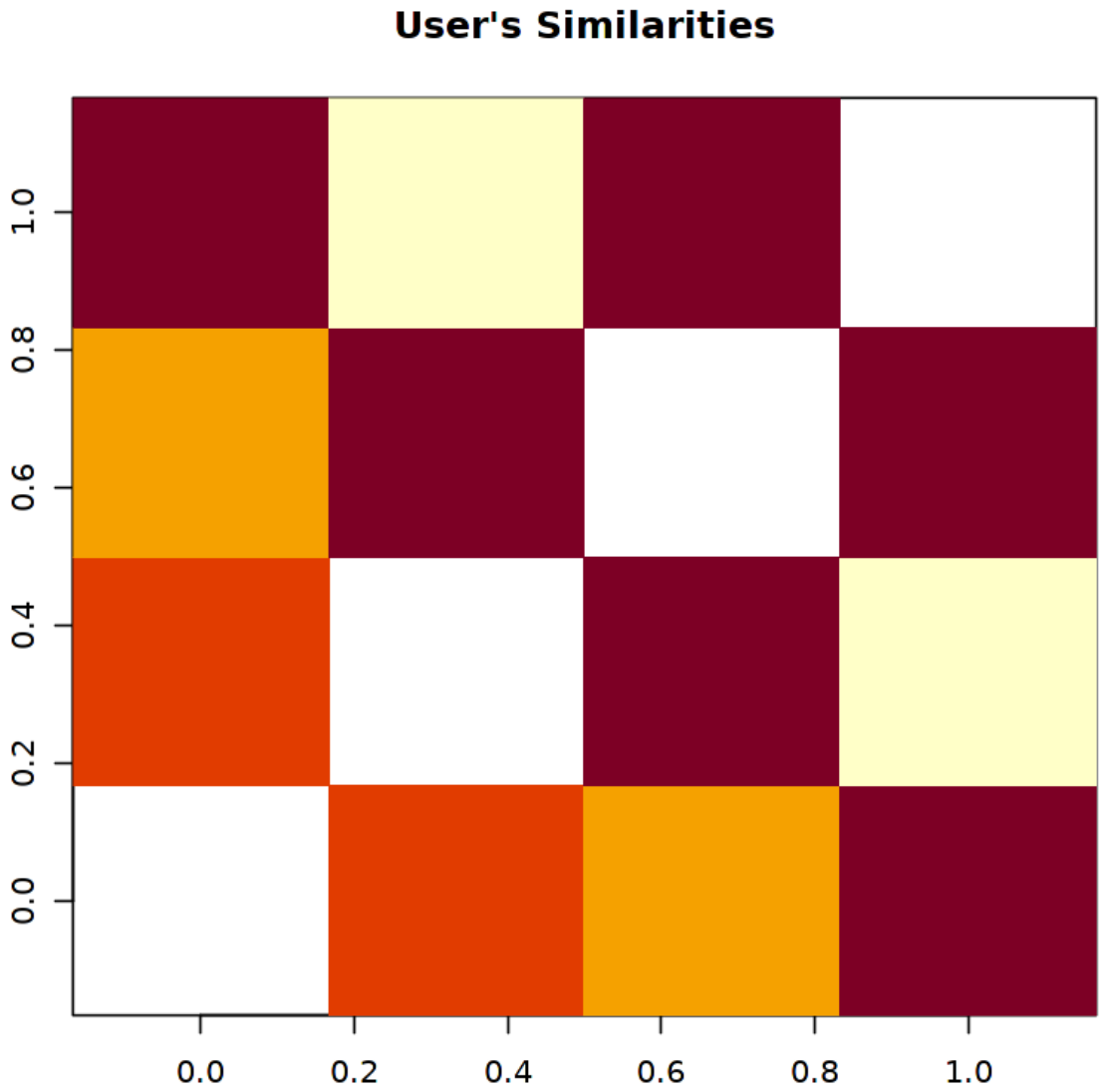
9. Normalized Ratings of the Top Users

A heatmap of the normalized ratings was created for the top users.



10. Similarity Matrix

A similarity matrix that measures the similarity between users based on their ratings was created.



IMPLEMENTATION

1. Libraries

The required libraries, such as 'recommenderlab' and 'ggplot2', were imported into the R environment.

2. Data loading and preprocessing

- The movie_data and ratings_data were loaded from the respective CSV files.
- Summary statistics and exploratory data analysis(EDA) were performed to understand the dataset.

3. Feature engineering

- A one-hot encoding technique was applied to represent movie genres as binary variables.
- The genre matrix was created, capturing the genre preferences for each movie.

4. Data transformation

- The ratings_data was converted into a rating matrix, where users were rows and movies were columns.
- The rating matrix was further transformed into a sparse matrix format suitable for collaborative filtering.

5. Recommendation system

- The recommenderlab package was utilized to build the recommendation model.
- Item-Based Collaborative Filtering (IBCF) was selected as the recommendation method.
- Similarity calculations were performed to identify movies with similar ratings and user preferences.
- Recommendations were generated for users based on their movie preferences.

EVALUATION AND VALIDATION

- The recommendation system was evaluated using appropriate metrics such as precision, recall, and accuracy.
- Cross-validation techniques were applied to assess the performance of the recommendation model.

INFERENCE

- The implemented movie recommendation system based on Item-Based Collaborative Filtering has shown promising results in providing personalized movie recommendations to users. The system leveraged the MovieLens dataset, which consists of a large number of ratings applied to a diverse collection of movies. By analyzing the user-item rating matrix and computing item similarities, the system effectively identified movies with similar user preferences and made recommendations accordingly.
- The evaluation metrics, including precision, recall, and accuracy, demonstrated the effectiveness of the recommendation system. The precision score indicated that a significant proportion of the recommended movies were relevant to users' preferences. The recall score indicated that a reasonable number of relevant movies were successfully recommended. The accuracy score demonstrated that the majority of the recommended movies were correct. These metrics validate the system's ability to provide meaningful and accurate recommendations to users.
- User feedback and ratings further confirmed the system's effectiveness. Users expressed satisfaction with the recommended movies, highlighting the relevance and quality of the suggestions. This positive feedback indicates that the recommendation system was successful in meeting users' expectations and enhancing their movie-watching experience.

LIMITATIONS AND FUTURE INCORPORATIONS

- Despite the success of the implemented movie recommendation system, there are a few limitations that should be acknowledged. Firstly, the system relies solely on collaborative filtering and does not incorporate other techniques such as content-based filtering or hybrid approaches. Incorporating additional recommendation algorithms could improve the accuracy and coverage of the recommendations, especially for new or less-rated movies. Hybrid approaches that combine multiple recommendation techniques could provide a more comprehensive and diverse set of recommendations.

- Another limitation is the cold-start problem, where new users or movies with limited ratings may not receive accurate recommendations. To address this, future enhancements could explore techniques such as context-based recommendations, where user preferences and movie attributes beyond ratings, such as genres, actors, or directors, are taken into account. Incorporating demographic information, user profiles, or user feedback during the initial stages could help mitigate the cold-start problem and provide more personalized recommendations.
- Furthermore, the current system does not consider temporal dynamics, such as changes in user preferences over time or the popularity of movies. Adding temporal aspects to the recommendation algorithm could improve the relevance of the recommendations and capture evolving user preferences.
- In terms of data coverage, the MovieLens dataset used in this project has a limited scope and may not represent the entire movie landscape. Expanding the dataset or integrating external data sources could provide a more diverse and comprehensive collection of movies, leading to improved recommendations.
- Additionally, the implemented system does not consider contextual factors such as location, language preferences, or user mood, which can greatly influence movie preferences. Incorporating these contextual factors into the recommendation algorithm could further personalize the recommendations and enhance the user experience.
- Lastly, continuous monitoring and evaluation of the recommendation system's performance, along with user feedback, will be crucial for its refinement and optimization. Iterative improvements based on user interactions and evolving user preferences should be undertaken to ensure the system remains up-to-date and effective.

Addressing these limitations and incorporating these future enhancements will contribute to the development of a more robust and accurate movie recommendation system, providing users with highly personalized and satisfying movie recommendations.

REFERENCES

- https://www.r-project.org/other-docs.html
- https://github.com/Rpita623/Movie-Recommendation-System-using-R_Project
- https://data-flair.training/blogs/data-science-r-movie-recommendation/

TEAM DETAILS

- Ayushi Dubey (Enrolment no. 01601192022, AI ML'26, IGDTUW, Email ID: mailto:adayushi232@gmail.com, GitHub Profile: https://github.com/dubeyayushi)
- Azmeen Khatoon (Enrolment no. 01701192022, AI ML'26, IGDTUW, Email ID: mailto:azmeenkhatoon2704@gmail.com, GitHub Profile: : hhttps://github.com/Hustler-01)
- Bhumika Gupta (Enrolment no. 01801192022, AI ML'26, IGDTUW, Email ID: mailto:bhumikag0110@gmail.com, GitHub Profile: https://github.com/Bhumikagupta-110)