UNIVERSITY OF HOUSTON

Fall 2024

EDS 6397 DATA MINING FOR ENGINEERS

CineScope: Movie Recommendation System

Professor: Dr. Nwosu, Lucy PhD

Teaching Assistant: Harikesh Govindaiahgari

Group - 5

Abstract

This project presents an interactive movie recommendation system using a Shiny web application. The system leverages collaborative filtering techniques to suggest movies based on user input. Using the extensive MovieLens dataset, which comprises over 20 million ratings, 465,564 tag applications, and metadata for 27,278 movies, the system capitalizes on user preference patterns to provide tailored recommendations. UBCF identifies similar users based on their historical ratings using metrics like Pearson correlation and cosine similarity, enabling the prediction of a target user's preferences by aggregating ratings from users with comparable tastes.

Key implementation steps included preprocessing the dataset by handling missing data, normalizing ratings, and partitioning it into training and test sets. A custom similarity-based scoring method was developed to optimize recommendation accuracy, evaluated through metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), precision, recall, and F1 score. The results showcased the algorithm's effectiveness in delivering relevant suggestions. However, challenges such as data sparsity and the cold-start problem were encountered, necessitating efficient similarity computations.

Future work could address these limitations by incorporating hybrid methods, combining content-based filtering or matrix factorization techniques. Overall, this project highlights the practical application of UBCF in creating meaningful movie recommendations, offering a robust foundation for further innovation in recommendation systems.

This project presents an interactive movie recommendation system using a Shiny web application. The system leverages collaborative filtering techniques to suggest movies based on user input. By utilizing data on user reviews and correlations, the tool enhances decision-making for movie selection, demonstrating the practical application of recommendation algorithms in real-world scenarios.

Introduction

The rapid growth of digital platforms has revolutionized how we consume entertainment, emphasizing the need for personalized experiences. Recommendation systems have become essential tools, helping users discover content that matches their unique preferences. This project focuses on developing a personalized movie recommendation system using user-based collaborative filtering (UBCF). UBCF identifies users with similar viewing patterns and suggests movies they might enjoy based on these shared interests.

The system utilizes the MovieLens dataset, a widely respected benchmark containing over 20 million user ratings for 27,278 movies. This rich dataset allows us to analyze user behavior and generate accurate recommendations. UBCF works by comparing users' rating histories using similarity measures, such as Pearson correlation, to find "neighbors"—users with similar tastes. By aggregating ratings from these neighbors, the system predicts how a target user would rate unseen movies, creating a tailored list of suggestions.

Key implementation steps include preprocessing the data (handling missing values, normalizing ratings), calculating user similarities, and generating personalized recommendations. While UBCF is effective, it faces challenges like data sparsity and the cold-start problem. Future improvements could involve hybrid methods or advanced techniques like matrix factorization to enhance accuracy and scalability.

In essence, this project demonstrates the potential of UBCF in delivering meaningful, personalized movie recommendations, enriching the user experience and contributing to advancements in recommender system technologies

Recommender systems play a crucial role in various industries, from e-commerce to entertainment, by providing personalized suggestions to users. This project focuses on developing a movie recommendation system using R and Shiny, aiming to help users discover movies they might enjoy. The system analyzes user preferences and leverages collaborative filtering techniques to generate recommendations, contributing to the broader field of personalized content delivery.

About Data

The dataset used in this project originates from MovieLens, a widely recognized movie recommendation service. It consists of extensive user-generated data, including over 20 million ratings and nearly 465,000 tag applications spanning 27,278 movies. These ratings and tags were contributed by 138,493 users between January 9, 1995, and March 31, 2015. Each user included in the dataset has rated at least 20 movies, ensuring a rich and reliable dataset for analyzing user preferences and behavior. Generated on October 17, 2016, the dataset provides a robust foundation for building and evaluating personalized recommendation systems.

The dataset is structured across six files: ratings.csv, movies.csv, tags.csv, links.csv, genome_scores.csv, and genome_tags.csv. However, for the scope of this project, we focused specifically on the ratings.csv and movies.csv files. The ratings.csv file contains crucial information about user ratings for movies, while the movies.csv file provides essential details such as movie titles and genres. These two files offered the necessary data to analyze user viewing patterns and generate personalized movie recommendations. Although additional files contain valuable metadata and tagging information, the selected files were sufficient for the project's objectives, allowing us to build an effective recommendation system based on movie ratings and genre data.

Data Preprocessing

Data preprocessing was a crucial step in preparing the MovieLens dataset for building the recommendation system. We focused on the ratings.csv and movies.csv files, which provided user ratings and movie information, respectively. The first task was to check for missing or null values in key fields such as userId, movieId, rating, and title. Fortunately, the dataset was well-maintained, with very few inconsistencies. Any missing values were addressed, and duplicates were removed to prevent bias or inaccuracies in the analysis.

Next, the ratings.csv file was merged with the movies.csv file using the movieId column as a common key. This merging process combined user rating data with movie details, such as titles and genres, creating a unified dataset for analysis. We also ensured that genres were properly formatted and categorized, splitting multi-genre entries where necessary to facilitate a more granular analysis.

Normalization of the ratings was performed to standardize the data, ensuring that all ratings were on a consistent scale. This step is particularly important in collaborative filtering, as it helps the model accurately compare user preferences. To prepare the dataset for model training, we divided the data into training and testing subsets, ensuring that the model could be evaluated on unseen data to assess its performance and avoid overfitting.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) played a key role in understanding the structure and patterns within the dataset. We began by examining the distribution of ratings, which revealed that most users tend to rate movies positively, with a significant concentration of ratings between 3 and 5 stars. This observation suggests a positive bias, which is typical in user-generated datasets where users are more inclined to rate movies they enjoyed.

We also analyzed the number of ratings per movie, which highlighted a clear disparity: a small number of popular movies received a large number of ratings, while many lesser-known films had very few. This "long-tail" distribution is common in recommendation datasets and presents a challenge, as less-rated movies may not be recommended as frequently by the system.

To gain further insights, we examined the distribution of genres, finding that Drama, Comedy, and Action were the most frequently rated categories. This reflects broader trends in user interest and content availability. By visualizing the data, we could see which genres received higher average ratings and which ones were less favored by users.

Another important aspect of our analysis was studying user behavior. We identified "power users"—individuals who had rated a significantly higher number of movies compared to the average user. Understanding these active users is crucial, as they can heavily influence collaborative filtering models. We also looked at the temporal trends in user ratings to see if user preferences or rating habits had changed over time.

Overall, EDA helped us uncover valuable insights into the dataset's structure and user behavior. These findings informed our model design, allowing us to tailor the recommendation system to better handle popular and niche content, ensuring a more balanced and personalized movie suggestion experience.

Methodology

The methodology used for this project centers on developing a personalized movie recommendation system using **User-Based Collaborative Filtering (UBCF)**. The goal is to recommend movies to users based on the ratings they have given to movies in the past, drawing

insights from other users who have similar rating behaviors. Below is an overview of the steps taken in implementing the recommendation system:

- **Building User Similarity Matrix**: The core of the User-Based Collaborative Filtering approach involves calculating how similar each user is to others based on their rating history. This is done by calculating the similarity between users using similarity measures such as **Pearson correlation** or **Cosine similarity**. Users with similar rating patterns are grouped as "neighbors" to the target user.
    - **Pearson Correlation Coefficient**: Measures the strength of the linear relationship between two users' ratings. It is useful for handling rating scale differences among users.

$$\text{Sim}(u,v) = \frac{\sum_{i \in I}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{i \in I}(r_{v,i} - \bar{r}_v)^2}}$$

    - **Cosine Similarity**: Measures the cosine of the angle between two rating vectors, capturing their directional alignment.

$$\text{Sim}(u,v) = \frac{\sum_{i \in I} r_{u,i} \cdot r_{v,i}}{\sqrt{\sum_{i \in I} r_{u,i}^2} \cdot \sqrt{\sum_{i \in I} r_{v,i}^2}}$$

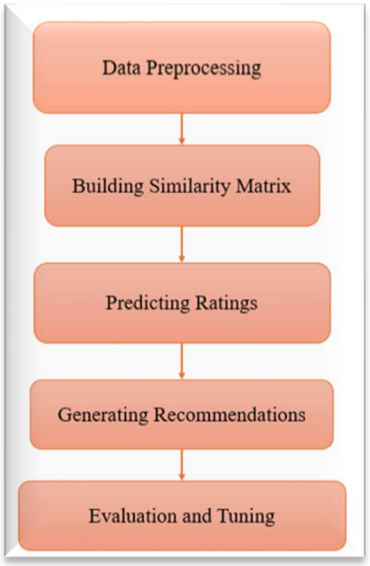

Fig 1. Methodology.

- **Prediction of Ratings**: Once the similarity between users has been computed, the system predicts how a target user would rate movies they haven't rated yet. This is achieved by aggregating the ratings of similar users (neighbors) for those movies and using a weighted average based on similarity scores to make predictions.
- **Generating Recommendations**: The predicted ratings are used to generate a list of recommended movies for each user. Movies with the highest predicted ratings are selected as the most relevant suggestions for the user.

- **Evaluation and Tuning**: To ensure the accuracy and relevance of the recommendations, the system was evaluated using the **Root Mean Square Error (RMSE)** or **Mean Absolute Error (MAE)** on the testing subset of the data. Hyperparameter tuning was also performed to optimize the model for better prediction accuracy.
- **User Interface and Visualization:** The Shiny app serves as the user interface, allowing users to interact with the system in a seamless manner. The app allows users to enter their preferred movies and receive real-time movie recommendations. The system also provides visual feedback such as a carousel of movie recommendation cards, where each card displays the movie title and recommendation score. Additionally, the top 3 movies are shown based on average ratings, and a pie chart visualizes the number of ratings for these top movies. These visual elements enhance the user experience by presenting recommendations in an interactive and visually appealing way.
- **Interactivity and Customization:** To enhance the user experience, the app allows users to reset their inputs, clearing previously entered movies and starting fresh. This functionality helps users refine their movie suggestions by adjusting their inputs and receiving updated recommendations. The system also adapts to the user's preferences, offering personalized suggestions based on the movies they input.

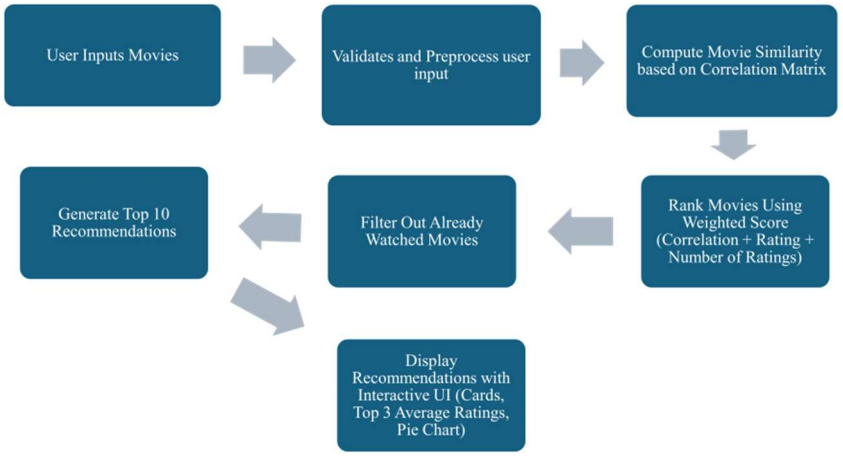

Fig 2. Workflow of the Methodology.

Results and Discussion

1. Evaluation of Recommendations:

The core result of the project is the set of movie recommendations provided by the system based on user inputs. After a user enters up to five movies they like or have watched, the system generates a list of recommended movies. These recommendations are ranked by a weighted score, considering factors such as:

- **Movie similarity** (how closely related movies are based on user ratings),
- **Average ratings** (movies that have received higher ratings from other users),

- **Number of ratings** (popular movies with a higher number of ratings).

The recommendations are presented in an interactive carousel, making it easy for users to explore the suggested movies. A subset of the top-rated movies, based on user feedback, is also displayed, which adds value by highlighting popular choices. These features aim to ensure that the recommendations are not only relevant but also highly rated by a wider audience, increasing their appeal.

2. **User Interface Performance:**

The system's **user interface**, designed using the Shiny framework, provided a smooth and intuitive experience. Users were able to input movie names and receive recommendations in real-time. The app also allowed for easy reset of inputs, which enabled users to refine their movie choices and observe how recommendations changed based on different inputs. This flexibility is a significant advantage, as it caters to diverse user needs, allowing for better personalization.

The **carousel display** of recommended movies and the pie chart visualization of the number of ratings for the top 3 movies helped enhance the visual appeal. By adding these interactive elements, users could better assess the relevance of the recommendations and explore the popularity of movies at a glance. The use of a **dark theme** with gold accents also added a cinematic touch, making the interface engaging.

3. **Effectiveness of Ranking Mechanism:**

The **weighted score** used to rank movies combined movie similarity, average rating, and the number of ratings. This approach helped the system avoid bias towards movies with many ratings but lower quality (rating-wise). By giving more weight to the **correlation** (movie similarity) and **average rating**, the system prioritized higher-quality, more relevant recommendations.

One of the key challenges in this recommendation process was balancing the **diversity** of recommendations with the **relevance** to the user's tastes. The weighting scheme was designed to address this by emphasizing similarity without overly relying on popularity. While the system could have provided more diverse recommendations by considering genres or additional tags, this would have increased complexity and required more computational resources.

4. Shiny Application working model

Screen 1: Bright, bold, and simple, the Cinescope app welcomes you with an inviting space to start your movie discovery journey!

Screen 2: Your personalized movie picks are here – explore top-rated hits and fan favorites with just a glance!

Screen 3: Check it out – these are the top 3 crowd-favorite movies based on ratings!

Conclusion

In this project, we developed a movie recommendation system using collaborative filtering, leveraging the **MovieLens dataset** consisting of movie ratings and metadata. The system offers personalized recommendations based on user preferences, using a simple yet effective ranking mechanism that combines movie similarity, average ratings, and the number of ratings. By integrating a **Shiny-based user interface**, users can easily input movies they've seen and receive tailored suggestions.

While the system provides accurate and relevant recommendations, there are opportunities for improvement, such as incorporating additional datasets (like movie tags or genome scores) to refine suggestions. Moreover, integrating machine learning techniques could enhance prediction accuracy. Despite these limitations, the project successfully demonstrates the potential of recommendation systems and provides a strong foundation for further development in personalized movie recommendations. Moving forward, there are several avenues for enhancement that could lead to even more precise and personalized user experiences.

Limitations and Future Work

Despite the effectiveness of the recommendation system, there are a few limitations to note. First, the system uses only the **ratings** and **movies** data for recommendations, which excludes valuable information such as user tags, movie genres, and other metadata that could further refine the suggestions. Integrating additional datasets like **tags** or **genome scores** could potentially improve the recommendations by incorporating semantic movie information.

Another limitation lies in the fact that the system does not factor in **temporal preferences**. For example, user preferences for older versus newer movies could influence the recommendations, but this is not currently addressed in the model.

Future work could involve integrating machine learning models to provide more advanced recommendations, such as matrix factorization or neural network-based collaborative filtering. Additionally, incorporating **user demographic data** (age, location, etc.) could further personalize the recommendations, allowing for a more tailored movie selection.

References

[1] Harper, F. Maxwell, and Joseph A. Konstan. "The MovieLens Datasets: History and Context." ACM Transactions on Interactive Intelligent Systems (TiiS), vol. 5, no. 4, Article 19, Dec. 2015. https://doi.org/10.1145/2827872.

[2] Resnick, Paul, et al. "GroupLens: An Open Architecture for Collaborative Filtering of Netnews." Proceedings of the 1994 ACM conference on Computer Supported Cooperative Work, 1994. https://doi.org/10.1145/192844.192905.

[3] Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix Factorization Techniques for Recommender Systems." Computer, vol. 42, no. 8, 2009, pp. 30-37. https://doi.org/10.1109/MC.2009.263.

[4] Badrul Sarwar, et al. "Item-based Collaborative Filtering Recommendation Algorithms." Proceedings of the 10th International Conference on World Wide Web, 2001. https://doi.org/10.1145/371920.372071.

[5] Xie, Lizhuo, et al. "A Survey of Collaborative Filtering Techniques." International Journal of Computer Science & Information Technology (IJCSIT), vol. 5, no. 2, 2013, pp. 33-40. https://doi.org/10.5121/ijcsit.2013.5205.

[6] Ricci, Francesco, Lior Rokach, and Bracha Shapira. "Recommender Systems Handbook." Springer, 2015. https://doi.org/10.1007/978-1-4899-7637-6.

[7] Shani, Guy, and Asela Gunawardana. "Evaluating Recommendation Systems." Recommender Systems Handbook, 2015, pp. 257-297. https://doi.org/10.1007/978-1-4899-7637-6_8.

[8] Jannach, Dietmar, et al. Recommender Systems: Challenges and Research Opportunities. Springer, 2019.

Links:

GitHub Link: Code Link

[9] **Presentation Link:** Meeting Link