A User-Based Collaborative Filtering Approach to Movie Recommendation

Jared Cox and Deesha Rajiv

The University of West Florida, 11000 University Parkway, Pensacola, FL, USA
{jac192,dr148}@students.uwf.com

1 Introduction

Recommender systems are a category of intelligent algorithms designed to suggest items to users based on various forms of input data, such as user preferences, past behaviors, and similarities to other users. These systems are commonly used in many real-world applications where personalization is essential. One of the most recognizable examples is in the domain of entertainment platforms like Netflix, Hulu, Disney+, and Max, where recommendation systems play a crucial role in improving user experience. These platforms analyze and study what a user may have previously watched, liked, or rated. With this data, they compare it with other users' preferences and the system can suggest new recommendations on movies and TV shows that the user is most likely to enjoy. This created a more engaging and exciting user experience that keeps users on the platform longer and helps them discover new content they might have otherwise missed.

For our project, we focused on developing and implementing a user-based collaborative filtering [2] recommendation system. This type of recommendation algorithm is centered around the idea that users who have agreed in the past are likely to agree in the future. Our main goal of this system was to predict and recommend movies for a specific user by identifying other users with similar views and rating patterns. To complete this, we utilized the Pearson Correlation [1] coefficient as our similarity metric, allowing us to measure how closely related two users' preferences are based on their common ratings.

The specific problem that we tackled involved processing a large dataset of user-movie interactions and transforming the data into recommendations. We had to parse, clean, and analyze thousands of individual user ratings to computer user similarities and generate personalized movie recommendations. This project highlights the real-world applicability of AI in creating smart, user-centered solutions that improve how people interact with digital content platforms.

2 Methods

To solve this problem, we implemented a collaborative filtering recommendation system that relies on the Pearson correlation coefficient. This works by taking the users' rated movies, finding users who have rated similarly in the past, and then using those ratings to predict ratings for movies the target user has not yet seen. For a pair of users A and B, we calculate similarity as

$$sim(A,\ B) \ = \ [\Sigma\,(rA,i \ - \ \bar{r}A)(rB,i \ - \ \bar{r}B)] \,/\, [\sqrt{\Sigma}(rA,i \ - \ \bar{r}A)^2 \times \sqrt{\Sigma}(rB,i \ - \ \bar{r}B)^2] \quad (1)$$

Once the similarity is calculated, we predict the target user's rating for a movie based on the weighted average rating of similar users.

$$\hat{r}A,j \ = \ \Sigma\,[sim(A,\ B) \times rB,j] \,/\, \Sigma\,|sim(A,\ B)| \qquad (2)$$

3 Experiments

To evaluate the accuracy and overall performance of our user-based collaborative filtering recommender system, we conducted a series of experiments using a large-scale data set comprised of movie ratings. The original data set contained more than 25 million individual entries, representing a unique instance of a user rating a particular movie. Due to the complexity of the dataset, we realized that using the entire dataset would be impractical and very computationally intensive for our testing and evaluation.

To simplify the dataset, we developed and executed a Python script that allowed us to create a smaller, more manageable subset of the data. The revised dataset consists of approximately 100,000 entries, which we found to be much more suitable for testing while still obtaining a realistic simulation of rating trends. Upon further exploration, we came to the conclusion that 100,000 entries were still too sizable to reliably conduct testing on. We simplified the dataset even further, with this iteration of the dataset having only 247 data entries. This simplified dataset allows us to quickly and reliably test our recommender system, while still obtaining accurate results.
The data is formatted as a CSV (Comma Separated Values) file, with each row consisting of three distinct columns: UserId, MovieName, and Rating. Each row represents a single rating that a specific user has given to a particular movie. This allows us to track user preferences and evaluate our recommender system's ability to generate personalized and accurate recommendations.

We employed the train-test split strategy to assess the accuracy of our collaborative-filtering-based recommender system. This strategy provides an effective method of simulating real-world scenarios by partitioning the dataset into training and testing sets. The data was divided such that our training set consisted of 80% of the total user-movie ratings. The purpose of this training set is to construct and train the recommender model by calculating user preferences and similarities. The remaining 20% of data is held back as the test set. This test set is selected randomly, and it contains user ratings that are used exclusively for the evaluation of the recommender system's performance. For each user in our dataset, 20% of their ratings are held back at random. The recommender system trained itself on the remaining 80% of ratings and attempted to predict the ratings from the test set based on what it learned from the training subset. To measure any discrepancy between the actual rating from the test subset, and the predicted rating from the recommender model, we calculated the average absolute error.

$$AAE = (1/n) \sum |\hat{r}_i - r_i| \quad (3)$$

This metric computes the average magnitude of the prediction errors without taking direction into account, therefore providing a reliable estimate of model accuracy.

To further experiment with our collaborative-filtering-based recommender system, we implemented Leave-One-Out Cross-Validation (LOOCV). This is a robust technique used to evaluate and assess the predictive performance of our recommender system. In this method, for each user, there is one rating that is held back from the dataset as a test. The user's remaining ratings are used to train the model. This process is repeated for every single rating in the dataset for each user, so every rating in the dataset is used as a test instance exactly once. This method gives us an incredibly comprehensive evaluation of the system as each prediction is validated independently, and it maximizes the set of training data. Like the train-test split method, we calculated the average absolute error [3] for each prediction to give us a nice, summary statistic of how close the recommender system's predictions were.

4 Results & Conclusions

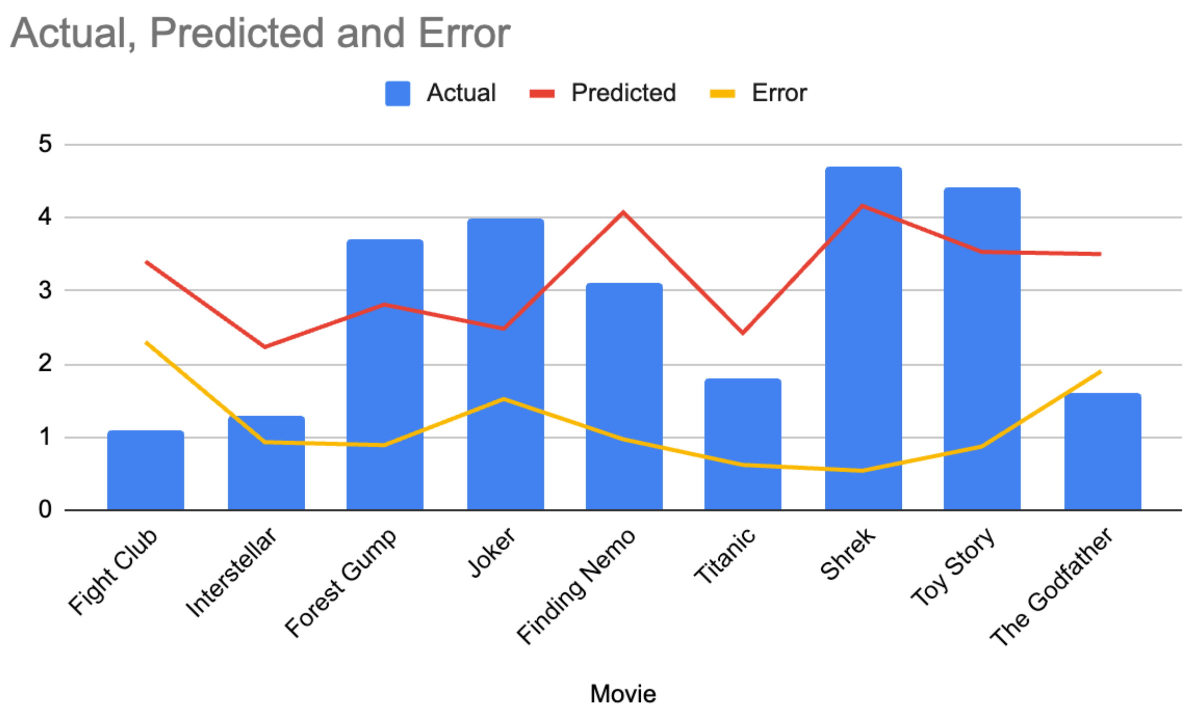

Fig. 1. Sample prediction results from the user-based collaborative filtering system. The table shows the actual and predicted ratings alongside the absolute error. These examples illustrate both accurate predictions and a few notable deviations.

As shown in Figure 1, the system's predictions are generally close to the actual ratings, with many errors falling below 1.0. However, there are a few outliers-such as *Fight Club and The Godfather*-that show higher error values. These deviations may have occurred due to inconsistent data or the system not being able to find users that closely match those preferences.

Across the entire test set, the average absolute error was 1.10. This indicates that our predicted ratings differ from the actual user ratings by just over one point on a five-point scale. This overall result suggests that the user-based collaborative filtering approach is reasonably effective in capturing user preference. While many of the predictions were accurate, there were still a few outliers that showed a notable deviation, highlighting the difficulty of predicting uncommon preferences. Overall, the system performed consistently in generating personalized movie recommendations.

Table 1. Ratings provided by User 12. These form the basis for identifying similar users and generating recommendations.

Movie Title	User Rating
The Matrix	2
Inception	4
Titanic	3
The Godfather	5
Interstellar	1

Table 2. Top 5 movie recommendations for User 12 based on their ratings of selected movies. Predictions were generated using user-based collaborative filtering and represent the expected user preference for unseen movies.

Rank	Movie Title	Predicted Rating
1	*The Dark Knight*	3.76
2	*La La Land*	3.53
3	*The Lion King*	3.40
4	*Toy Story*	3.34
5	*The Social Network*	3.09

Based on the data presented in Table 1 and the recommendations in Table 2, our user-based collaborative filtering method was reasonably effective at generating relevant movie suggestions. The system recommended popular movies such as *The Dark Night* and *Toy Story,* which align with the genres and styles of movies the user rated highly such as *The Godfather and Inception*, for example.

The predictions appear to reflect the general preferences of users with similar rating patterns, suggesting that the Pearson correlation-based similarity measure worked how we expected it to. However, some of the predicted ratings fall within a range of about 3.0-3.7 which may suggest that the system tends to make cautious predictions or has difficulty distinguishing between highly preferred and moderately liked movies.

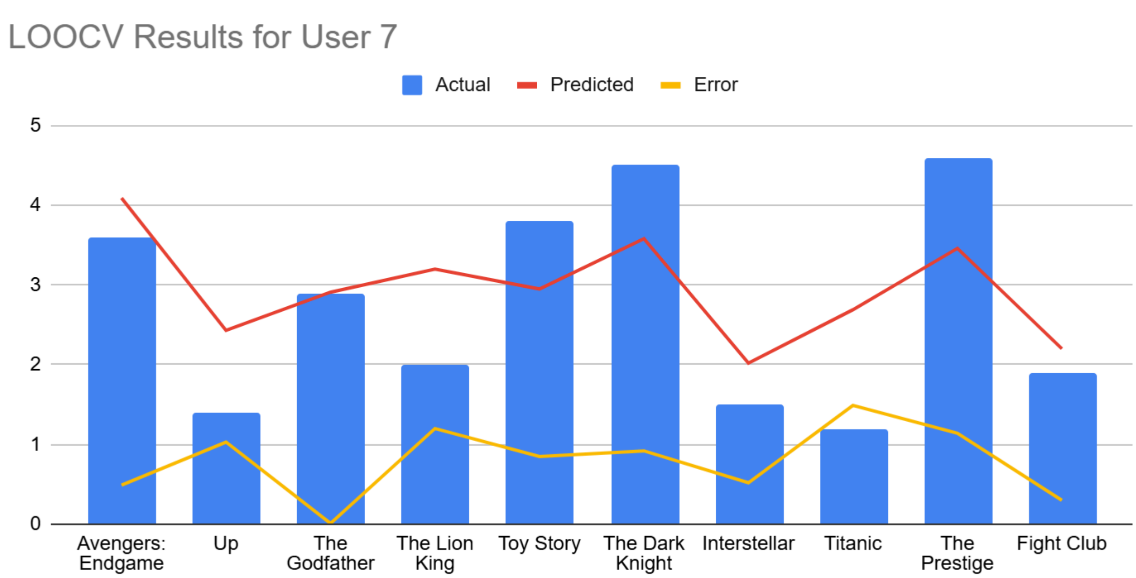

Fig. 2. Results from the Leave-One-Out Cross-Validation experiment for user 7. This graph shows the actual rating compared with the prediction from the LOOCV test. The error for each prediction is shown.

As shown in Figure 2, the LOOCV experiment resulted in predictions that closely approximated user 7's actual ratings for most of the movies with several predictions landing within a margin of 1.0 and one prediction being only producing an error of 0.01. There are outliers, just like we saw with the train-test split experiment, but overall the data that we procured shows that our system was able to reasonably capture and predict the user's movie preferences.

Overall, the method met our expectations in terms of functionality and baseline accuracy, especially for a relatively simple implementation. The recommender system was able to generate personalized predictions that aligned reasonably well with user preference, demonstrating that user-based collaborative filtering using Pearson correlation is an effective approach for this type of problem. However, there is still room for improvement. One potential improvement is to weigh each neighbor's contribution based on their similarity score. This would allow more similar users to have a stronger influence on the final prediction. Another way we could improve the system is to include additional data sources, this could include movie genres, release years, or user demographics. These improvements would not only enhance user satisfaction but also increase both the accuracy and personalization of future recommendations.

References

1. Berman, J.J., *Data Simplification*, Morgan Kaufmann (2016).
2. Bhatnagar, Vishal: *Collaborative Filtering Using Data Mining and Analysis*. Information Science Reference, Hershey, Pennsylvania (2017).
3. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning, Second Edition: Data Mining, Inference, and Prediction*. Springer (2009)