# A Comparison of Movie Recommendation Algorithms

Eddie McGowan

Lehigh University, Bethlehem, PA 18015, USA

**Abstract.** This paper evaluates the following five movie recommendation algorithms using the MovieLens dataset - random selection, most-rated, highest-rated, collaborative filtering, and alternating least squares (ALS). The models' recommendations are compared using both quantitative and qualitative metrics. Quantitatively, the models' performance was assessed using mean absolute error (MAE) and area under the precision-recall curve (AUC-PR) across varying recommendation thresholds. Qualitatively, the models' recommendations were evaluated based on their ability to cater to users' preferences and suggest high-quality/novel movies. The ALS model consistently outperformed the others, achieving the lowest MAE and highest AUC-PR, likely due to leveraging the user's complete rating history for personalized recommendations. Collaborative filtering demonstrated strong qualitative performance in identifying similar movies but struggled on quantitative metrics, likely due to its limited access to a user's full preference history. Both the highest-rated and most-rated models excelled at recommending widely-liked and critically-acclaimed films but lacked novelty. The random selection model served as a baseline, exhibiting the weakest performance overall. These findings highlight the strengths and trade-offs of different recommendation approaches, emphasizing the importance of aligning model design with application goals and user preferences. Future work could explore hybrid approaches, combining the strengths of multiple models to enhance recommendation quality and user satisfaction.

**Keywords:** Recommender Systems · ALS · Collaborative Filtering

## 1 Introduction

Recommendation algorithm suggest items to users based on specific criteria and are widely used across industries, including e-commerce, social media, and streaming platforms. During week 4 of this class, we built a music recommendation algorithm using the Audioscrobbler dataset and an alternating least squares (ALS) approach. ALS, initially developed by researchers at HP Labs for the Netflix Prize, continues to be one of the most widely adopted collaborative filtering algorithms for large-scale recommendation systems [1, 2]. For this project, ALS was compared against four other recommendation algorithms.

In this project, the MovieLens dataset is used , and five recommendation models were tested: a randomly selected movie model, a most-rated movie model, a highest-rated movie model, a cosine similarity item-based collaborative filtering model, and an ALS model [3, 4]. Each model allows users to filter recommendations by genre. To test real-world applicability, the models were evaluated using three criteria: mean absolute error (MAE), area under the precision-recall curve, and results for my movie preferences.

### 1.1 Movie Lens Dataset

MovieLens is a website that recommends movies to users based on their ratings of past movies. MovieLens releases an anonymized version of their data via GroupLens Research for recommendation system research. GroupLens is a research lab in the Department of Computer Science and Engineering at the University of Minnesota [3, 4].

The dataset contains 87,585 movies and 32,000,204 movie ratings by 200,948 users between January 09, 1995 and October 12, 2023. This dataset is the most comprehensive and up-to-date movie rating dataset publicly available at the time of this paper. This dataset was generated on October 13, 2023, and released publicly in May 2024. The dataset contains four data files. First, a movie dataset with the movie's ID, title, and genre. Second, a rating dataset that contains an anonymized UserID and the corresponding movieID and rating for the movie they are reviewing. Ratings varied between 0.5 to 5 in increments of 0.5. Third, a link dataset links each movie to its corresponding IMDB and The Movie Database page. Lastly, a tags dataset that provides a set of words or a sentence corresponding to each movie [4].

## 2 EDA

### 2.1 Data Loading

The movies and ratings datasets were used for the analysis, while the tags and links datasets were excluded as irrelevant to the project's scope. The data was preprocessed to ensure no duplicates, with ID columns loaded as integers, ratings as doubles, and titles/genres as strings. Users had one rating per movie they watched, and the movies and ratings datasets were joined to associate movie titles and genres with ratings for the recommendation algorithms.

### 2.2 Genre Analysis

The dataset includes genres such as action, comedy, drama, and sci-fi, among others (see Fig. 1). Movies can belong to multiple genres. While a few movies lacked genre information, most films missing genres were short video clips rather than major films, so this missing data should have minimal impact on the recommender system (see Fig. 2).

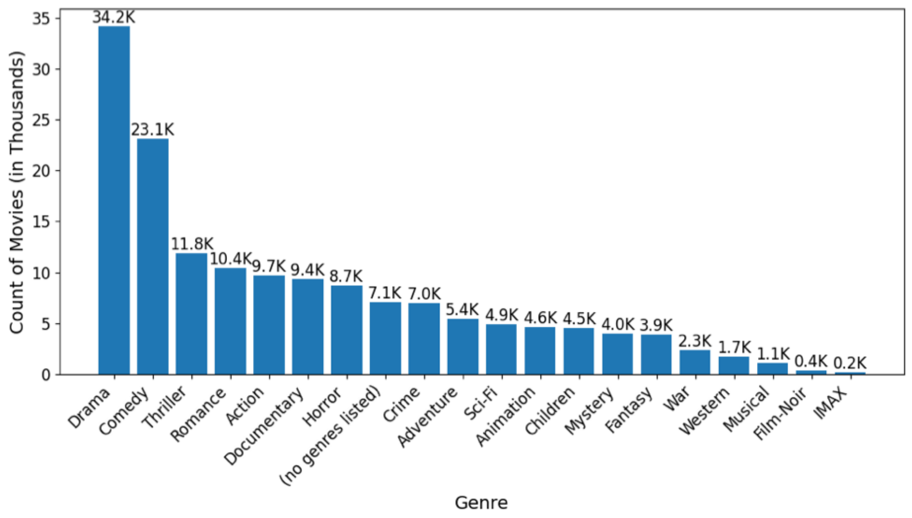
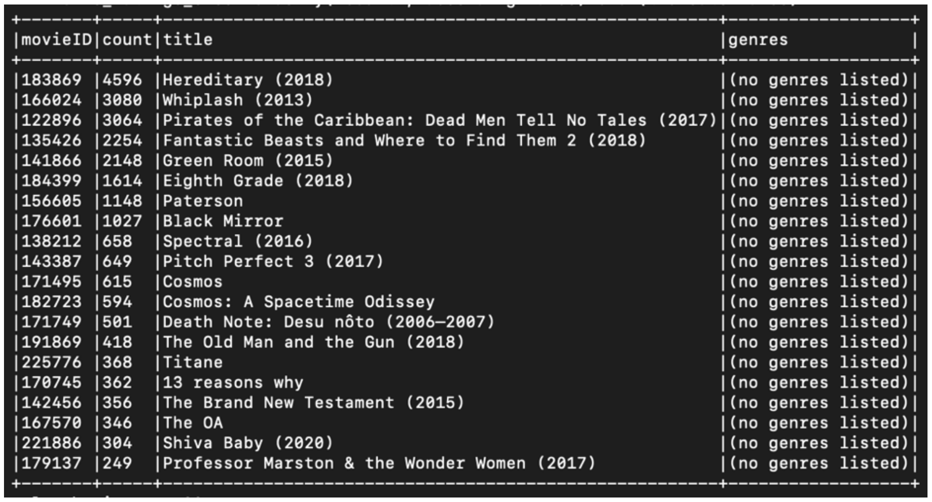


**Fig. 1.** Bar chart of the number of movies by genre.



```
+-------+-----+------------------------------------------------------+-------------------+
|movieID|count|title                                                 |genres             |
+-------+-----+------------------------------------------------------+-------------------+
|183869 |4596 |Hereditary (2018)                                     |(no genres listed)|
|166024 |3080 |Whiplash (2013)                                       |(no genres listed)|
|122896 |3064 |Pirates of the Caribbean: Dead Men Tell No Tales (2017)|(no genres listed)|
|135426 |2254 |Fantastic Beasts and Where to Find Them 2 (2018)      |(no genres listed)|
|141866 |2148 |Green Room (2015)                                     |(no genres listed)|
|184399 |1614 |Eighth Grade (2018)                                   |(no genres listed)|
|156605 |1148 |Paterson                                              |(no genres listed)|
|176601 |1027 |Black Mirror                                          |(no genres listed)|
|138212 |658  |Spectral (2016)                                       |(no genres listed)|
|143387 |649  |Pitch Perfect 3 (2017)                                |(no genres listed)|
|171495 |615  |Cosmos                                                |(no genres listed)|
|182723 |594  |Cosmos: A Spacetime Odissey                           |(no genres listed)|
|171749 |501  |Death Note: Desu nôto (2006–2007)                     |(no genres listed)|
|191869 |418  |The Old Man and the Gun (2018)                        |(no genres listed)|
|225776 |368  |Titane                                                |(no genres listed)|
|170745 |362  |13 reasons why                                        |(no genres listed)|
|142456 |356  |The Brand New Testament (2015)                        |(no genres listed)|
|167570 |346  |The OA                                                |(no genres listed)|
|221886 |304  |Shiva Baby (2020)                                     |(no genres listed)|
|179137 |249  |Professor Marston & the Wonder Women (2017)           |(no genres listed)|
+-------+-----+------------------------------------------------------+-------------------+
```

**Fig. 2.** Most-rated movies with no genre listed.

### 2.3 User analysis

A histogram of the number of movies watched by users is shown in Fig. 3. All of the users had a minimum of 20 movie ratings. The mean number of movie ratings by users was about 159, whereas the median was 73. This difference between mean and median was due to a heavy right tail caused by a subset of highly active users. The distribution of their ratings can be seen in Fig. 4. The average movie rating was about 3.5.

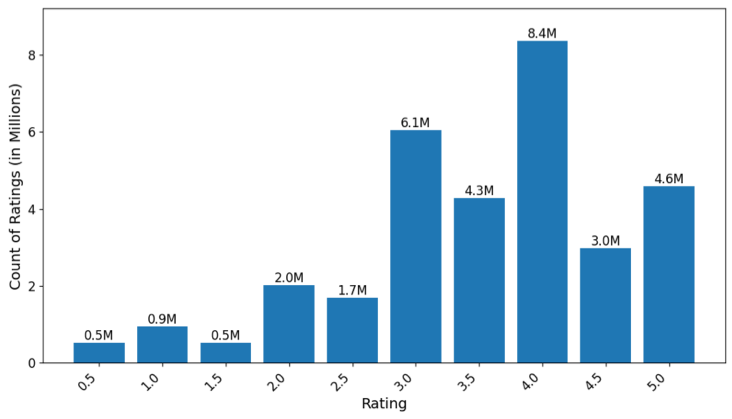


**Fig. 3.** Histogram of the number of ratings by rating score.

## 3 Recommendation Systems and Qualitative Results

The following section provides a detailed description of the five recommendation systems. Each model was provided with a complete list of movies, their corresponding ratings across the dataset, and the movies that the user receiving the recommendations has watched along with the ratings they assigned. This setup ensured that all movies were available to the model at the start and prevented the model from recommending movies the user had already seen. By default, each model generates ten recommendations, but this number can be adjusted to suit individual needs. In addition, each model includes a genre filter, allowing users to tailor recommendations based on their genre preferences. The models also predicted user ratings, which were compared to the user's actual ratings to evaluate their quantitative performance.

Numerous recommendation systems have been developed using the MovieLens datasets. While the following papers served as sources of inspiration, no code from these works was directly utilized in this project [5, 6].

To qualitatively evaluate these models, I rated 200 movies and added myself to the dataset. Below, I analyze each model's results for my data based on

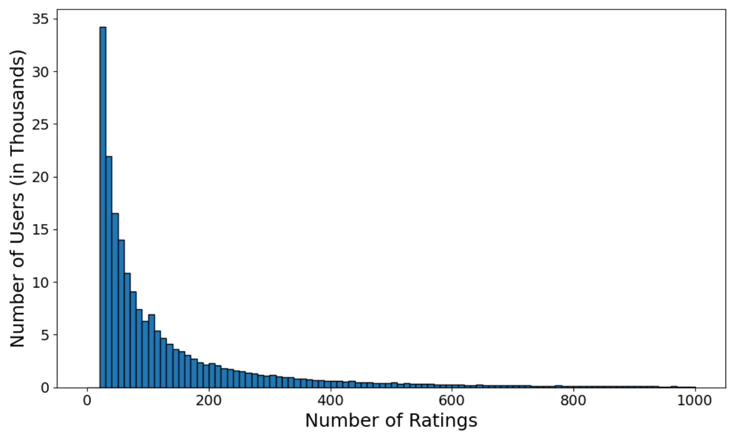


**Fig. 4.** Histogram of the distribution of the number of movies rated by users (bin size: 10 movies).

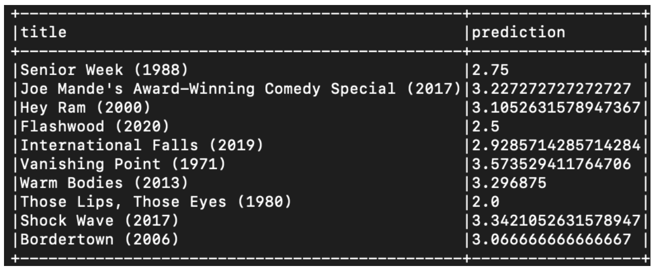
two criteria: first, the relevance of the recommendations to my preference for critically acclaimed dramas and action thrillers; and second, the number of high quality movies recommended by the model that I had not previously considered watching. At the end of this section, I will rank the models based on the quality of their recommendations.

### 3.1 Randomly Selected Movie (See Fig. 5)

This model recommends a random movie that the user has not yet seen. To ensure repeatability, a random seed is generated based on the user ID, allowing recommendations to remain consistent for a given user while varying across users. The model served as a baseline for comparison with more advanced models. For each recommended movie, the average rating is displayed as the predicted score, representing the likely rating the user might assign to that movie. This model did not personalize its recommendations based on user preferences.

After researching each of the recommended movies in Fig. 5, I found that this model performed poorly in aligning with my interests. Only three out of the ten films appealed to me *Vanishing Point, Shock Wave, and Bordertown*. Of these, I would only consider watching *Vanishing Point* as it is the only movie rated above the average user rating above 3.5.

However, the model excels at recommending films I had not previously considered watching. Of the ten movies suggested, I was familiar with only one - *Warm Bodies*. While the recommendations were not tailored to my preferences or of consistently high quality, the model's strength lied in uncovering niche films that might otherwise be overlooked by other recommendation systems.



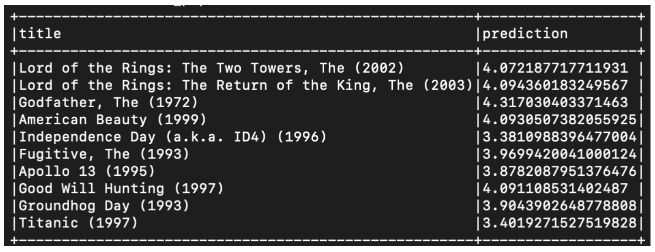
```
+-----------------------------------------------+------------------+
|title                                          |prediction        |
+-----------------------------------------------+------------------+
|Senior Week (1988)                             |2.75              |
|Joe Mande's Award-Winning Comedy Special (2017)|3.227272727272727 |
|Hey Ram (2000)                                 |3.1052631578947367|
|Flashwood (2020)                               |2.5               |
|International Falls (2019)                     |2.9285714285714284|
|Vanishing Point (1971)                         |3.573529411764706 |
|Warm Bodies (2013)                             |3.296875          |
|Those Lips, Those Eyes (1980)                  |2.0               |
|Shock Wave (2017)                              |3.3421052631578947|
|Bordertown (2006)                              |3.066666666666667 |
+-----------------------------------------------+------------------+
```

**Fig. 5.** Randomly selected movie model results for the author.

## 3.2 Most-Rated (See Fig. 6)

This model recommends the most-reviewed movies that the user has not yet seen. For each recommendation, the average rating is displayed as the predicted score, representing the likely rating the user might assign to that movie. As shown in Fig. 6, most of the recommended movies were blockbusters or "classic" films, frequently accompanied by high average ratings. This result was likely because users who review movies on MovieLens tend to gravitate toward widely acclaimed English-language films. This model does not personalize its recommendations to user preferences.

The model aligned well with my interests, as all the recommendations were critically acclaimed dramas or action thrillers. I enjoyed watching most, but not all, of the films I had seen from this list. However, this model was less effective at suggesting films I have not previously considered watching. This model, however, may be particularly beneficial for younger users or international users who may not be familiar with these culturally significant English-language films. Still, the model's recommendations lacked novelty, as a high percentage of Americans having already seen most of these movies. Further, for those movies that remain unseen, many users likely have intentionally chosen not to watch these well-known movies. For instance, I have been repeatedly recommended to watch *American Beauty*, but I have avoided it despite its alignment with my genre preferences because I am not interested in its subject matter.



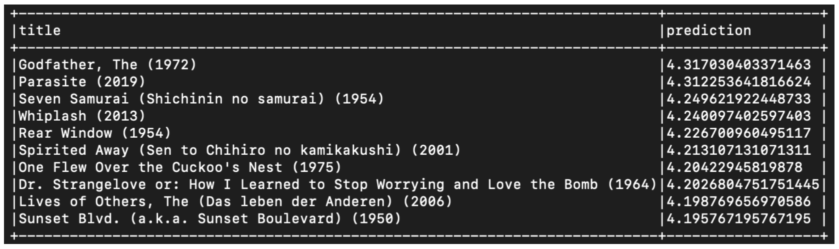
```
+-----------------------------------------------------+------------------+
|title                                                |prediction        |
+-----------------------------------------------------+------------------+
|Lord of the Rings: The Two Towers, The (2002)        |4.072187717711931 |
|Lord of the Rings: The Return of the King, The (2003)|4.094360183249567 |
|Godfather, The (1972)                                |4.317030403371463 |
|American Beauty (1999)                               |4.0930507382055925|
|Independence Day (a.k.a. ID4) (1996)                 |3.3810988396477004|
|Fugitive, The (1993)                                 |3.9699420041000124|
|Apollo 13 (1995)                                     |3.8782087951376476|
|Good Will Hunting (1997)                             |4.091108531402487 |
|Groundhog Day (1993)                                 |3.9043902648778808|
|Titanic (1997)                                       |3.4019271527519828|
+-----------------------------------------------------+------------------+
```

**Fig. 6.** Most-rated movie model results for the author.

Overall, this model excels at recommending critically acclaimed and widely liked movies, often aligning with user interests. However, it is less effective at helping users discover new films, which limits its appeal for those seeking novel recommendations. As discussed in Section 4, this model is likely to perform well on quantitative metrics such as MAE and area under the precision-recall curve due to the high quality and broad appeal of its recommendations. Nevertheless, these metrics may overstate the model's effectiveness, as it fails to introduce most users to new and exciting films aligned with their preferences.

## 3.3 Highest-Rated (See Fig. 7)

This model recommends the movies with the highest average rating that the user has not yet seen. For each recommendation, the average rating is displayed as the predicted score, representing the likely rating the user might assign to that movie. To ensure quality recommendations, I set a minimum threshold of 3,000 ratings for each movie. This threshold was introduced after observing that some movies with very few ratings were achieving a perfect 5 out of 5 score. These were often not traditional films but rather student projects and commercials, whose high ratings were likely influenced by close connections to the film (e.g., cast, friends, or family). By incrementally increasing the threshold, I ensured that the recommendations aligned with well-known top movie lists, such as IMDb's Top 250 movie list [7]. This model also does not personalize its recommendations to user preferences.



```
+---------------------------------------------------------------------------+------------------+
|title                                                                      |prediction        |
+---------------------------------------------------------------------------+------------------+
|Godfather, The (1972)                                                      |4.317030403371463 |
|Parasite (2019)                                                            |4.312253641816624 |
|Seven Samurai (Shichinin no samurai) (1954)                                |4.249621922448733 |
|Whiplash (2013)                                                            |4.240097402597403 |
|Rear Window (1954)                                                         |4.226700960495117 |
|Spirited Away (Sen to Chihiro no kamikakushi) (2001)                       |4.213107131071311 |
|One Flew Over the Cuckoo's Nest (1975)                                     |4.20422945819878  |
|Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb (1964)|4.2026804751751445|
|Lives of Others, The (Das leben der Anderen) (2006)                        |4.198769656970586 |
|Sunset Blvd. (a.k.a. Sunset Boulevard) (1950)                              |4.195767195767195 |
+---------------------------------------------------------------------------+------------------+
```

**Fig. 7.** Highest-rated movie model results for the author.

This model performed well for me, as all the films in Fig. 7 were aligned with my movie interests. I enjoyed all of the films I had seen from this list. Out of the recommendations, I had only not considered watching three of these movies, which is an improvement over the most-rated model. These new films were foreign masterpieces, such as *Seven Samurai*, and Golden Age films, such as *Sunset Blvd.* In addition, the model also uncovered lesser-known movies, such as *The Lives of Others*. Lastly, the model still recommended many widely recognized classics, such as *The Godfather* or *Rear Window*. Overall, this diversity in film styles makes the model more relevant to a broader audience than the most-rated model. However, despite this improvement, many recommendations still were not particularly novel or surprising as they frequently appeared on curated top movie lists such as IMDb's Top 250 [7]. Thus, users who haven't seen their recommended movies may likely have previously decided against watching them due to disinterest. For example, I have repeatedly been recommended to watch *Sunset Blvd.*, but have avoided watching it because I am not interested in its subject matter. As discussed in Section 4, similar to the most-rated model, this model is likely to perform well on quantitative metrics such as MAE and area under the precision-recall curve due to the high quality of its recommendations. Nevertheless, these metrics may overstate the model's effectiveness, as it is still relatively poor at introducing users to new and exciting films aligned with their preferences.

## 3.4 Item-based Collaborative Filtering (See Fig. 9)

This model recommends movies that a user has not yet seen based on the movies they have previously rated highly. Cosine similarity is used to identify similarities between films, enabling the model to infer that if a user enjoys movie X, they are likely to enjoy a similar movie Y. This is the most complex of the models created, and for clarity, a flowchart of the model is included in Fig. 8.

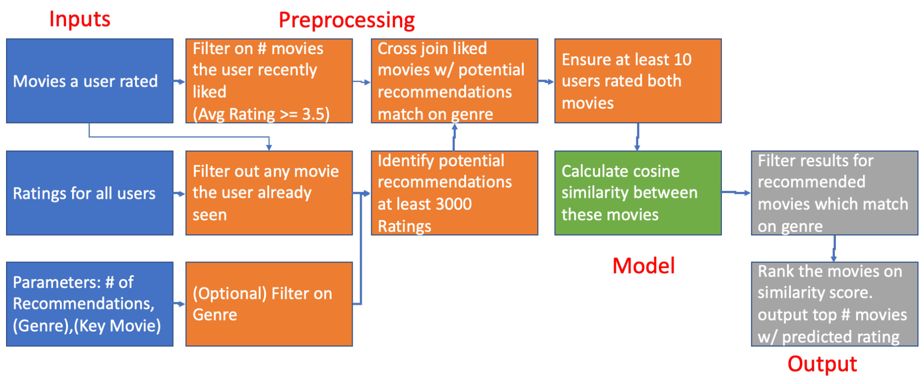

Inputs
Preprocessing
Movies a user rated
Filter on # movies the user recently liked (Avg Rating >= 3.5)
Cross join liked movies w/ potential recommendations match on genre
Ensure at least 10 users rated both movies
Ratings for all users
Filter out any movie the user already seen
Identify potential recommendations at least 3000 Ratings
Calculate cosine similarity between these movies
Filter results for recommended movies which match on genre
Parameters: # of Recommendations, (Genre),(Key Movie)
(Optional) Filter on Genre
Model
Rank the movies on similarity score. output top # movies w/ predicted rating
Output

**Fig. 8.** Flowchart of the item-based collaborative filtering model.

To determine which movies a user enjoys, their ratings are filtered to include only they rated 3.5 or higher. This threshold corresponds to the average user rating in the dataset, ensuring that only their above-average movies are selected. In parallel, a table of all movies the user has not seen is created, containing the ratings each user has assigned to these movies. Similar to the highest-rated model, this table is filtered to only include movies with over 3,000 ratings, ensuring the focus remains on theatrical films. Without this filter, the model would include student films and short films, which are outside the interests of most users.

Due to the complexity of the model, only ten percent of the data in this table was used as a subset. This was necessary to prevent PySpark from overflowing. The original dataset contained 88 million movie ratings, so even with a 10 percent subset, millions of records remained for the model's calculations, ensuring the recommendations were still robust.

The table of movies the user enjoys is then cross-joined with the table of unseen movies. Cosine similarity is calculated between the overall ratings for movies the user enjoyed and the ratings for movies the user has not seen:

As shown in Equation 1, cosine similarity measures the angle between two vectors [8]. When applied to this dataset, the vector represents the ratings of two movies "A" and "B" by users who have rated both films. "i" refers to a particular user who has rated both films, and $r_{A_i}$ and $r_{A_i}$ refers to their ratings for movies "A" and "B" respectively. Cosine similarity values range from -1 (completely dissimilar) to 1 (completely similar) [8]. One potential issue is if two movies are rated by only one user and this user assigns the same rating to both films, then the cosine similarity would be a perfect 1. This issue was seen in earlier iterations of this model, so a threshold was set requiring at least 10 users to have rated both movies.

$$\text{Cosine Similarity} = \frac{\sum_{i=1}^{n} r_{A_i} \cdot r_{B_i}}{\sqrt{\sum_{i=1}^{n} r_{A_i}^2} \cdot \sqrt{\sum_{i=1}^{n} r_{B_i}^2}} \tag{1}$$

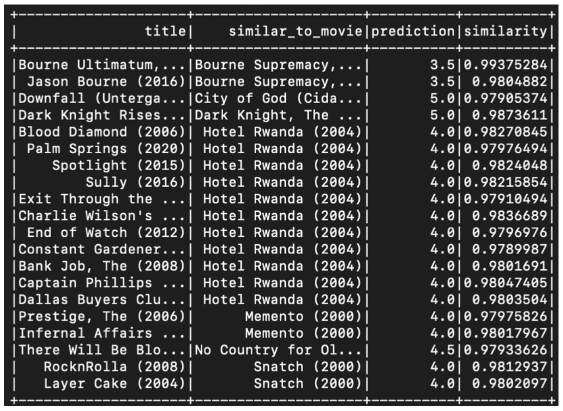
The computational complexity of the similarity calculation is $O(nr^2)$, where n is the number of users, and r is the average number of ratings per user [9]. For the full dataset, this would require approximately 4.5 billion computations (201K users *(159 average movies seen per user)$^2$). This high computational cost necessitated subsampling the data. Additionally, to avoid redundant computations, the cosine similarity between movies is cached.

After calculating cosine similarity, a genre filter was applied to ensure the recommended movie aligns with the genre of the positively rated movie. For example, without this filter, a user who enjoys the movie *The Godfather* might be recommended the movie *Moana* due to overlapping user preferences for critically acclaimed films, despite the stark difference in genre and tone between the two movies.

Finally, potential recommendations were ranked and recommended in descending order of similarity score. The recommended movie was displayed alongside the movie the user previously enjoyed. The predicted rating for the recommendation was set to the same rating the user gave the movie they enjoyed. For example, if a user rated *The Godfather* 4.5 stars, the model predicts they would also rate *The Godfather Part II* 4.5 stars. By default, recommendations were based on all movies the user has liked, which can sometimes result in recommendations spanning different tones and genres. To address this, the model included an option to recommend movies similar to a specific title, allowing users to tailor recommendations further.

This model performed well for my interests, as all the films in Fig. 9 align closely with my preferences. I have enjoyed almost every film I've seen from this list. Since the model identified and recommended a high number of movies similar to Hotel Rwanda, the number of recommendations was increased for this test to demonstrate the quality of the results. Notably, I had not previously considered watching half of the recommended movies, which is a significant improvement over the most-rated and highest-rated models.

Cosine similarity effectively recommends movies that aligns with user preferences. For example, it correctly associates sequels from *The Bourne* and *The Dark Knight* series, and links *Hotel Rwanda* with other intense dramas. Additionally, it matched movies by the same directors, such as *Snatch* with *RocknRolla* and *Memento* with *The Prestige*.



```
+--------------------+--------------------+----------+----------+
|               title|    similar_to_movie|prediction|similarity|
+--------------------+--------------------+----------+----------+
|Bourne Ultimatum,...|Bourne Supremacy,...|       3.5|0.99375284|
| Jason Bourne (2016)|Bourne Supremacy,...|       3.5| 0.9804882|
|Downfall (Unterga...|City of God (Cida...|       5.0|0.97905374|
|Dark Knight Rises...|Dark Knight, The ...|       5.0| 0.9873611|
|Blood Diamond (2006)| Hotel Rwanda (2004)|       4.0|0.98270845|
| Palm Springs (2020)| Hotel Rwanda (2004)|       4.0|0.97976494|
|    Spotlight (2015)| Hotel Rwanda (2004)|       4.0| 0.9824048|
|        Sully (2016)| Hotel Rwanda (2004)|       4.0|0.98215854|
|Exit Through the ...| Hotel Rwanda (2004)|       4.0|0.97910494|
|Charlie Wilson's ...| Hotel Rwanda (2004)|       4.0| 0.9836689|
| End of Watch (2012)| Hotel Rwanda (2004)|       4.0| 0.9796976|
|Constant Gardener...| Hotel Rwanda (2004)|       4.0| 0.9789987|
|Bank Job, The (2008)| Hotel Rwanda (2004)|       4.0| 0.9801691|
|Captain Phillips ...| Hotel Rwanda (2004)|       4.0|0.98047405|
|Dallas Buyers Clu...| Hotel Rwanda (2004)|       4.0| 0.9803504|
|Prestige, The (2006)|      Memento (2000)|       4.0|0.97975826|
|Infernal Affairs ...|      Memento (2000)|       4.0|0.98017967|
|There Will Be Blo...|No Country for Ol...|       4.5|0.97933626|
|   RocknRolla (2008)|       Snatch (2000)|       4.0| 0.9812937|
|   Layer Cake (2004)|       Snatch (2000)|       4.0| 0.9802097|
+--------------------+--------------------+----------+----------+
```

**Fig. 9.** Cosine similarity item-based collaborative filtering model results for the author.

Unlike the previous two models, many of the recommended movies were less culturally significant or critically acclaimed. This creates a trade-off between reward and risk, with the risk lying in the fact that these movies, while highly rated, tend to be more divisive than the recommendations from the earlier models. The reward, however, is the opportunity to discover new films that closely align with a user's specific interests. Personally, I find the reward outweighed the risk, as this is the first model tested that recommends multiple high-quality, novel movies that align with my preferences.

Thus, this model provides value to a wide-range of users but has some limitations. One major issue is the cold start problem: it requires users to have rated at least one movie to serve as a baseline for recommendations. Additionally, the model can infer false preferences for a particular subgenre. For example, *The Dark Knight* is one of my favorite movies, but I dislike most superhero films, including *The Dark Knight Rises*. My high rating of *The Dark Knight* might align my preferences with users who enjoy superhero movies, leading to recommendations for multiple superhero films that I would not enjoy. This potential issue is addressed in the following model.

### 3.5 ALS: Alternating Least Squares (See Fig. 11)

This model recommends movies that the user has not seen based on all the movies the user has rated. The model identifies latent factors in user-movie interactions to predict user preferences. Unlike item-based filtering, which computes similarities between movies directly, ALS (Alternating Least Squares) focuses on uncovering hidden patterns in the data through matrix factorization. This approach is more scalable and allows the model to better understand the user's overall interests. For clarity, a flowchart of the model is included in Fig. 10 [1, 2].

Similar to the previous models, this model filters for "popular" movies with at least 3,000 ratings, focusing on theatrical releases. The ALS algorithm is trained on all ratings from this filtered dataset, mapping users and their ratings to movies and identifying latent factors. Initial parameters were set to align with those used in the ALS model for the Audioscrobbler music recommendation algorithm. The following adjustments were made for this project [1, 2]:

A nonnegative parameter was added to ensure all predicted movie ratings were positive. The implicitPrefs parameter was set to false, as the movie ratings represent explicit user preferences. The maxIter parameter was increased from 5 to 10,increasing the number of possible training iterations. Thus, improving the quality of recommendations. The regParam parameter was reduced from 0.1 to 0.01, allowing the model to more closely fit the data [10, 11]. Once trained, the ALS model generates a list of potential recommendations for the user and assigns a predicted rating to each movie. Predicted ratings are constrained to values between 0.5 and 5.0. Finally, the recommendations are ranked in descending order of predicted ratings and presented to the user.

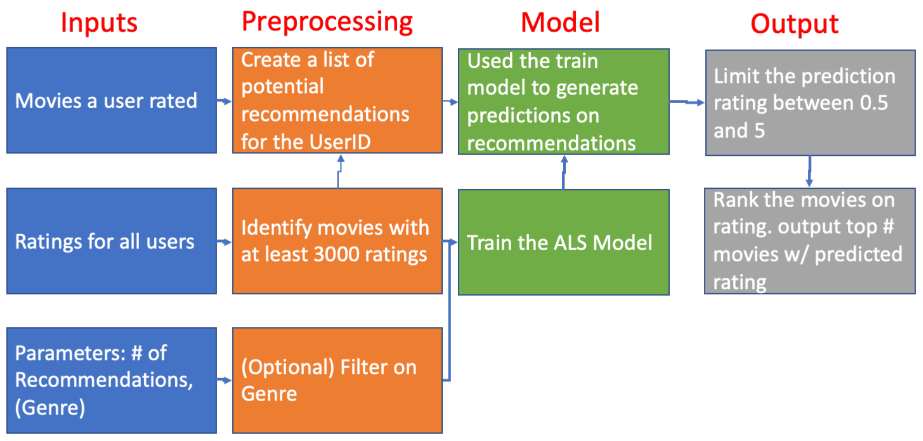
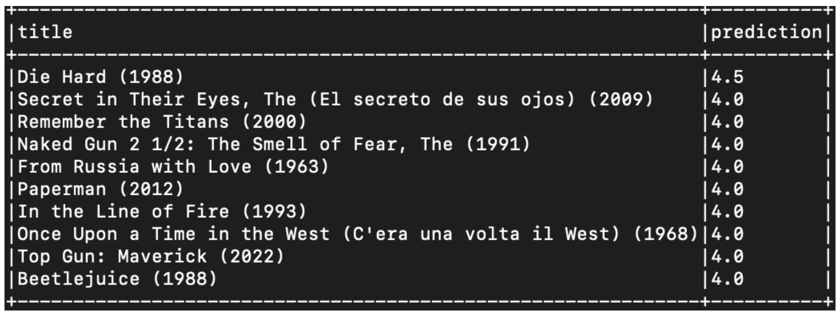

Inputs
Preprocessing
Model
Output
Movies a user rated
Create a list of potential recommendations for the UserID
Used the train model to generate predictions on recommendations
Limit the prediction rating between 0.5 and 5
Ratings for all users
Identify movies with at least 3000 ratings
Train the ALS Model
Rank the movies on rating. output top # movies w/ predicted rating
Parameters: # of Recommendations, (Genre)
(Optional) Filter on Genre

**Fig. 10.** Flowchart of the ALS model.

This model performed exceptionally well for my preferences, as all the films in Fig. 11 align closely with my tastes. I have enjoyed every movie I've seen from this list, most of which are action films or critically acclaimed dramas. Notably, unlike the recommendations from the other models, I had not previously considered watching the majority of the recommended movies.



```
+-------------------------------------------------------------+----------+
|title                                                        |prediction|
+-------------------------------------------------------------+----------+
|Die Hard (1988)                                              |4.5       |
|Secret in Their Eyes, The (El secreto de sus ojos) (2009)    |4.0       |
|Remember the Titans (2000)                                   |4.0       |
|Naked Gun 2 1/2: The Smell of Fear, The (1991)               |4.0       |
|From Russia with Love (1963)                                 |4.0       |
|Paperman (2012)                                              |4.0       |
|In the Line of Fire (1993)                                   |4.0       |
|Once Upon a Time in the West (C'era una volta il West) (1968)|4.0       |
|Top Gun: Maverick (2022)                                     |4.0       |
|Beetlejuice (1988)                                           |4.0       |
+-------------------------------------------------------------+----------+
```

**Fig. 11.** ALS model results for the author

Since this model uses all the movies a user has previously rated to generate recommendations, its results are generally more aligned with the user's true interests than the item-based collaborative filtering model, which focuses on one film at a time. As a result, the ALS model produced the best qualitative results of all the models tested. However, it, nevertheless, is not without flaws. For instance, it recommended the Disney short film *Paperman*. While critically acclaimed, short films fall outside my primary interests.

Overall, this model provides value to a wide range of users but is limited by the cold start problem in that it requires users to have rated at least one movie to serve as a baseline for recommendations.

### 3.6 Qualitative Comparison of Models

Qualitatively, the ALS model performed the best, closely followed by the collaborative filtering model. Both of these models successfully capture user preferences and recommend high-quality, lesser-known movies unlikely to appear in generic searches. The highest-rated model performed well but primarily recommends only widely recognized films. As these films are often part of the cultural zeitgeist, this model offered limited value to users who have already encountered or decided against watching them. The most-rated model, which similarly focuses on well-known films with limited novelty, performed the fourth best. Finally, the random selection model ranked the lowest of the five tested models. While its suggestions are new to most users, they are of significantly lower quality compared to the other models.

## 4 Quantitative Comparison of Models

To evaluate the discrepancy between each model's predictions and users' ratings, 5-fold cross-validation was used. This evaluation was conducted on a random subset of users with at least 100 ratings. The minimum threshold of 100 ratings ensured that the models had sufficient data in the test set to produce varied and reliable results. To complete the 2,500 model iterations required for these evaluations, the models were run overnight. Each model was trained on 80 percent of the filtered dataset (four folds), with the remaining 20 percent (the fifth fold) reserved for generating recommendations.

The models were evaluated at the following recommendation thresholds to evaluate overall performance : 3, 5, 7, 9, 11, 13, and 15. At each threshold, the metrics MAE (mean absolute error), precision, and recall were calculated. Additionally, each model's performance was visualized on a precision-recall curve, and the area under the precision-recall curve (AUC-PR) was computed.

As shown in Equation 2, MAE quantifies the average error of each model by measuring the absolute difference between the predicted rating (PR) and the true user rating (TR) across "n" recommended movies:

$$\text{MAE} = \frac{\sum_{i=1}^{n} |PR_i - TR_i|}{n} \tag{2}$$

The area under the precision-recall curve (AUC-PR) evaluates the model's ability to recommend movies that users are likely to enjoy, with a focus on maximizing the true positive rate. This metric is similar to the more commonly used area under the receiver operating characteristic (ROC) curve, as studied in Week 4 of this class [1]. However, unlike ROC AUC, AUC-PR does not account for true negatives. True negatives represent cases where the model correctly identifies movies that should not be recommended. These are less relevant in recommendation systems, where the primary goal is to identify and recommend a small subset of highly relevant movies from a large pool. In this context, true positives are far more critical than true negatives.

To calculate AUC-PR, precision and recall were computed at each recommendation threshold. Movies rated 3.5 or higher by the user were classified as positive recommendations, while those rated below 3.5 were classified as negative reviewed recommendations. Precision measures the percentage of positively reviewed films among all films in the recommendation set, as defined in Equation 3:

$$\text{Precision} = \frac{\text{Number of Positively Reviewed Recommendations}}{\text{Total Recommendations}} \tag{3}$$

Recall measures the percentage of positively reviewed films in the test set that are successfully included in the recommendations. as defined in Equation 4:

$$\text{Recall} = \frac{\text{Number of Positively Reviewed Recommendations}}{\text{Total Positively Reviewed Movies in Test Set}} \tag{4}$$

After calculating the precision and recall values, the results for each threshold are sorted by recall. Using the trapezoidal rule, the area under the curve is computed as shown in Equation 5. The sum of these areas across all intervals provides the AUC-PR value [12]:

$$\text{AUC-PR} = \sum_{i=1}^{n-1} \frac{\text{Precision}_i + \text{Precision}_{i+1}}{2} \cdot (\text{Recall}_{i+1} - \text{Recall}_i) \tag{5}$$

The model results for MAE and AUC-PR are summarized in Table 1, and the precision-recall curve is shown in Fig. 12. A lower MAE and a higher AUC-PR indicate better model performance.

**Table 1.** Comparison of Models: Mean Absolute Error (MAE) and Area under the Precision-Recall Curve (AUC PR). Tested for 100 users with at least 100 ratings.

| **Model** | **MAE** | **AUC-PR** |
|---|---|---|
| Random Model | 0.783 | 0.220 |
| Most-Rated Model | 0.672 | 0.288 |
| Highest-Rated Model | 0.659 | 0.307 |
| Collab Filtering Model | 0.641 | 0.265 |
| ALS Model | 0.400 | 0.332 |

Among the tested models, the ALS model performed the best on both MAE and AUC-PR. Its personalization, based on the user's entire rating history, gives it a significant advantage over the other models. The collaborative filtering model ranked second in MAE but fourth in AUC-PR, indicating strong performance in predicting user ratings but reduced effectiveness in identifying movies the user would enjoy. Initially, I hypothesized that this discrepancy was due to the model's parameters. To test this hypothesis, I raised the threshold for identifying liked movies in this model from 3.5 to 4. However, rerunning the analysis yielded similar results (MAE = 0.594, AUC-PR = 0.265), suggesting the is not the cause. As discussed in the qualitative modeling section, the primary limitation likely stems from inferred false preferences. The collaborative filtering model assumes user preferences based on individually liked movies, which can result in riskier recommendations. While this approach can often succeed, leading to a low MAE, occasional incorrect inferences can result in a significant number of errors, lowering AUC-PR compared to models that make safer predictions. The highest-rated model, achieved the third-best MAE and the second-best AUC-PR. As discussed in the qualitative modeling section, this model performs unexpectedly well because it consistently recommends critically acclaimed and widely liked movies, minimizing the risk of user dissatisfaction. A similar pattern was observed in the most-rated model, which also tends to recommend broadly popular films, leading to strong performance on these metrics. Lastly, the random model performed the worst on both MAE and AUC-PR, underscoring the superior performance of the other models compared to the baseline.

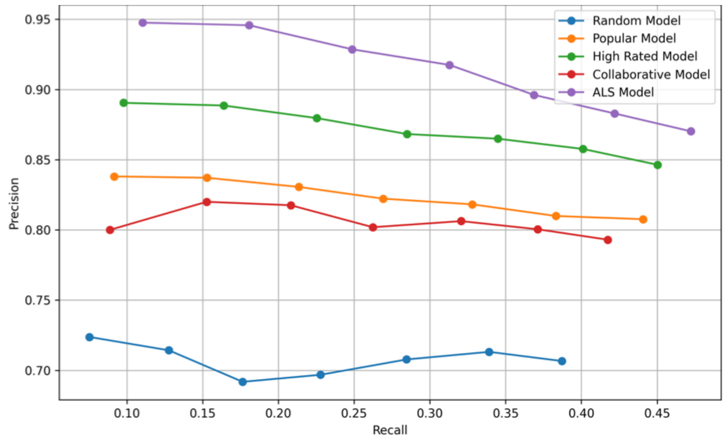


**Fig. 12.** Precision-recall curve for each model. Tested for 100 users with at least 100 ratings at thresholds of 3, 5, 7, 9, 11, 13, and 15 movie recommendations.

To determine whether the models' performance improves or declines as the number of user ratings increases, I compared the performance of models for users with between 100 and 200 ratings to their performance for users with between 200 and 300 ratings. In theory, there should be little to no change in the performance of the non-personalized models (random selection model, most-rated model, and highest-rated model). However, the personalized models are expected to improve as more data becomes available to generate personalized recommendations. MAE can be directly compared between the two tests. However, AUC-PR cannot be directly compared, as the recall is inherently lower in the 200-to-300 ratings set due to a larger number of positively reviewed movies in the test set (see Equation 4). Nonetheless, the ordering of the models based on AUC-PR can still be compared between the two sets.

The results are summarized in Table 2, and the corresponding precision-recall curves are shown in Fig. 13 and Fig. 14. The ordering of model performance by AUC-PR is consistent. I conducted a two-sample t-test to compare MAE between the two sets to determine if the difference in performance was statistically significant at a p-value threshold of 0.05. The p-value was above 0.05 for each of the five models, indicating insufficient evidence to prove a statistically significant difference in performance. There may be a plateau in model performance once users reach a certain number of ratings.

While this quantitative analysis provides insight into model performance, it is limited by the movies available in the test set. For example, if a user has not rated many niche movies, the collaborative filtering model may not fully demonstrate its potential value. If these models were implemented, they should initially be deployed to a subset of users to evaluate their actual performance.

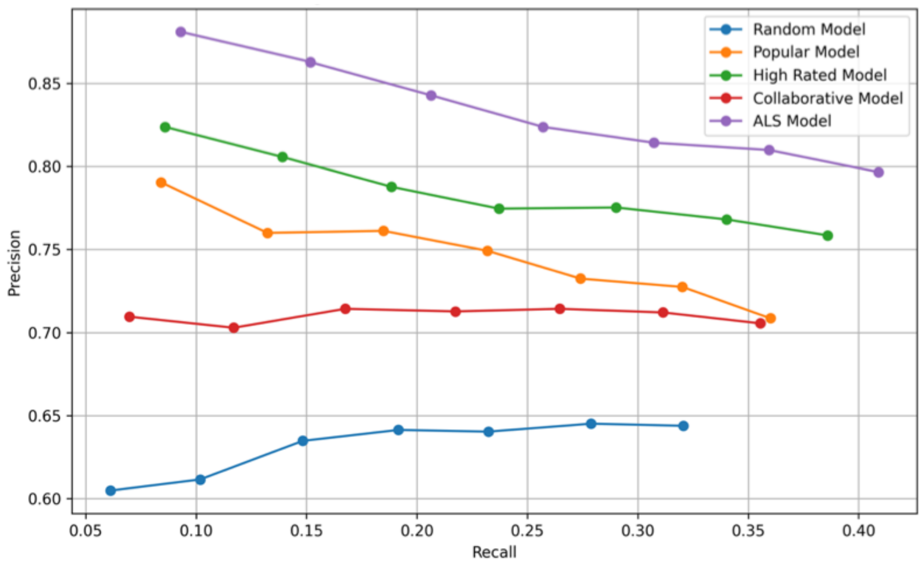


**Fig. 13.** Precision-recall curve for each model. Tested for 100 users with between 100 and 200 ratings at thresholds of 3, 5, 7, 9, 11, 13, and 15 movie recommendations.

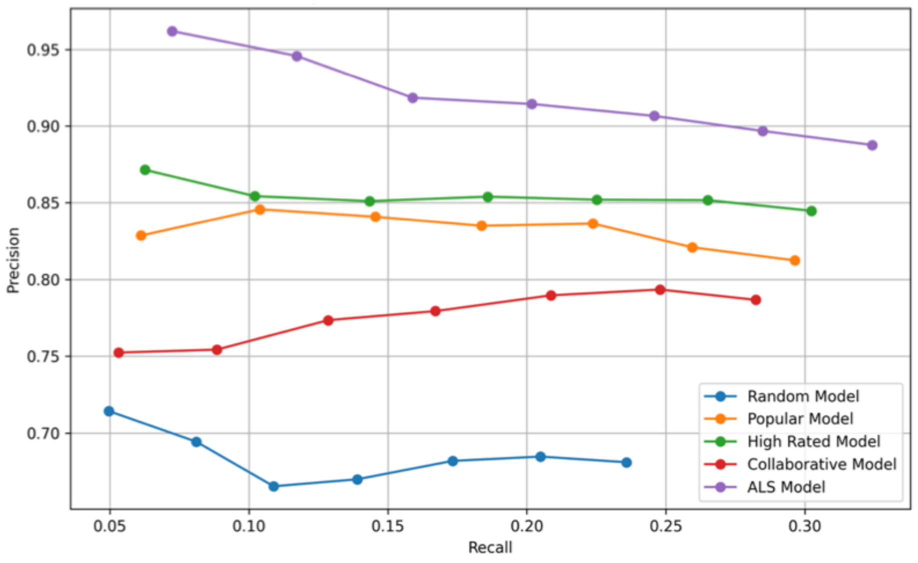


**Fig. 14.** Precision-recall curve for each model. Tested for 100 users with between 200 and 300 ratings at thresholds of 3, 5, 7, 9, 11, 13, and 15 movie recommendations.

**Table 2.** Comparison of Models: Mean Absolute Error (MAE) and Area under the Precision-Recall Curve (AUC-PR). Tested for 100 users with between 100 to 200 ratings (100-200) and 100 users with between 200-300 ratings (200-300).

| **Model** | **MAE (100-200)** | **AUC-PR (100-200)** | **MAE (200-300)** | **AUC-PR (200-300)** |
|---|---|---|---|---|
| Random Model | 0.720 | 0.332 | 0.700 | 0.127 |
| Most Rated Model | 0.698 | 0.370 | 0.733 | 0.196 |
| Highest Rated Model | 0.706 | 0.402 | 0.669 | 0.204 |
| Collab Filtering Model | 0.606 | 0.361 | 0.803 | 0.178 |
| ALS Model | 0.388 | 0.440 | 0.461 | 0.231 |

## 5 Further Study

With additional time and resources, a single-blind study where users watch and rate movies recommended by one of the models could be beneficial to determine the true model performance. Moreover, combining the strengths of different models through an hybrid approach may further enhance performance. For instance, latent factors from the ALS model could be used to generate recommendations, while features from the most-rated and highest-rated models could adjust rankings. Movies with a high number and percentage of recent ratings likely indicate new and popular content, whereas high average ratings paired with a substantial number of recent positive ratings suggest high-quality movies. Additionally, item-based collaborative filtering could be added as a factor to identify similar movies based on cosine similarity.

## 6 Conclusion

This paper evaluated the following five movie recommendation algorithms using the MovieLens dataset: random selection, most rated, highest rated, collaborative filtering, and ALS. These models were compared using both quantitative metrics, such as MAE and AUC-PR, and qualitative analysis of their recommendations.

The ALS model demonstrated the best overall performance, leveraging the user's complete rating history to deliver highly personalized recommendations. While the collaborative filtering model excelled at identifying similar movies, its limited access to the user's full preference history led to weaker performance on quantitative metrics. The highest-rated and most-rated models effectively recommended widely liked films but lacked novelty, while the random selection model, serving as a baseline, performed the worst in both quantitative and qualitative evaluations.

These findings underscore the strengths and trade-offs of each approach, emphasizing the need to align model design with specific application goals and user needs. Future work could explore combining the strengths of these models into an hybrid approach, which may yield a model with even better results.

## References

1. Tandon, A., Ryza, S., Laserson, U., Owen, S., & Wills, J. (2022). Chapter 3. Recommending Music and the Audioscrobbler Dataset. In *Advanced Analytics with PySpark: Patterns for Learning from Data at Scale Using Python and Spark*. O'Reilly Media. Retrieved from https://learning.oreilly.com/library/view/advanced-analytics-with/9781098103644/ch03.html#idm46507984812352
2. Zhou, Y., Wilkinson, D., Schreiber, R., & Pan, R. (2008). Large-scale parallel collaborative filtering for the Netflix Prize. *HP Labs.*
3. MovieLens. (n.d.). MovieLens: Personalized Movie Recommendations. MovieLens. Retrieved from https://movielens.org
4. MovieLens. (n.d.). MovieLens 32M Dataset. GroupLens Research. Retrieved from https://grouplens.org/datasets/movielens/
5. Rapaport, E. (n.d.). MovieLens 1M Deep Dive: Part I. *Towards Data Science*. Retrieved from https://towardsdatascience.com/movielens-1m-deep-dive-part-i-8acfeda1ad4
6. Gonçalves, C. (n.d.). Collaborative Filtering-Based Recommender System From Scratch. *Medium*. Retrieved from $https://medium.com/@camilolgon/collaborative-filtering-based-recommender-system-from-scratch-38037932b877$
7. IMDb Top 250 Movies. (2018). IMDb. *Medium*. Retrieved from https://www.imdb.com/chart/top/
8. Varun. Cosine similarity: How does it measure the similarity, Maths behind and usage in Python. *Medium; Towards Data Science.*. Retrieved from https://towardsdatascience.com/cosine-similarity-how-does-it-measure-the-similarity-maths-behind-and-usage-in-python-50ad30aad7db
9. Tian, D. (2021, September 12). Address Matching through Cosine Similarity. *Medium*. Retrieved from https://dt1086.medium.com/address-matching-through-cosine-similarity-f68f3b48c110
10. ALS — PySpark 3.5.3 documentation. (2024). *Apache.org.*. Retrieved from https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.recommendation.ALS.html
11. Borges, B. (2023, July 25). Recommender System using ALS in PySpark. *Medium*. Retrieved from $https://medium.com/@brunoborges_38708/recommender-system-using-als-in-pyspark-10329e1d1ee1$.
12. Precision-Recall. (2024). *Scikit-Learn.*. Retrieved from $https://scikit-learn.org/1.5/auto_examples/model_selection/plot_precision_recall.html$