Journal of Management and Service Science, 2024,
Vol. 04, Iss. 01, S. No. 062, pp. 1-14
ISSN (Online): 2583-1798

A Hybrid Approach to Movie Recommendation System

Abhay Yadav[1], Garima Srivastava[2], Sachin Kumar[3]

[1,2,3]Amity School of Engineering and Technology, Amity University, Lucknow Campus, India
[1]abhayyaduvanshi0111@gmail.com, [2]gsrivastava1@lko.amity.edu, [3]skumar3@lko.amity.edu

How to cite this paper: A. Yadav, G. Srivastava, S. Kumar, "A Hybrid Approach to Movie Recommendation System," *Journal of Management and Service Science (JMSS),* Vol. 04, Iss. 01, S. No. 062, pp. 1-14, 2024.

https://doi.org/10.54060/a2zjournals.jmss.62

Received: 09/06/2023
Accepted: 10/03/2024
Online First: 25/04/2024
Published: 25/04/2024

Abstract

Recommendation Systems (RS) have become indispensable in today's digital ecosystem, influencing decisions and experiences across several platforms. This paper delves deeply on RS, including its history, functionality, and relevance. It covers many aspects of RS, such as Content-Based and Collaborative Filtering, using examples from a variety of industries, including e-commerce and entertainment. The paper also describes and covers empirical analysis methodologies for comparing RS efficacy and providing a framework. By conducting a qualitative and quantitative analysis, compared these three recommendation systems i.e. content based collaborative and Hybrid. This mixed analysis approach was necessary as Content-Based Filtering systems are not easily quantifiable, and for a movie recommendation system, the qualitative aspect holds significant importance. Through our analysis, it became evident that a hybrid recommendation system consistently outperforms standalone methods in terms of recommendation accuracy and relevance.

Keywords

Recommendation Systems, Origins, Functionalities, Content-Based Filtering, Collaborative Filtering, Empirical Analysis, Effectiveness, Decision-making

1. Introduction

In the modern digital landscape, Recommendation Systems (RS) have become an integral part of our online interactions, influencing decisions and experiences across various platforms. This chapter aims to delve into the origins, functionalities, and significance of Recommendation Systems. Through a thorough exploration of different categories within RS and the provision of relatable examples, this chapter seeks to enhance understanding and provide valuable insights. Additionally, an empirical analysis will be conducted to compare the effectiveness of different recommendation system types, offering clear conclusions to aid decision-making. Finally, this chapter will outline the structure of the thesis, offering a roadmap for the subsequent chapters.

1.1. Origins of Recommendation Systems

Recommendation Systems have played a crucial role in guiding human decision-making processes. These systems analyze user preferences and behaviors to provide personalized suggestions, categorized primarily into Content-Based Filtering and Collaborative Filtering Recommendation Systems. While early recommendation systems relied on basic similarity-measure-based approaches, modern iterations incorporate advanced techniques such as Machine Learning and Deep Learning.

The success of recommendation systems spans diverse industries, including e-commerce, entertainment, and media. Notable platforms leverage recommendation engines to enhance user experiences and drive engagement. Table 1 showcases examples of popular platforms utilizing recommendation technology.

Table 1. Some notable platforms that Use Recommendation System

Site	What is recommended?
Amazon	Consumer Product
Netflix	Movies and TV Series
Facebook	Friend Suggestion
CareerBuilder	Jobs
YouTube	Videos
Tinder	Dates

1.2. What Does A Recommendation System Do?

According to the definition provided in [1], a recommendation system, also known as a recommender system, falls under the category of information filtering systems. Its primary objective is to anticipate the rating or preference a user might assign to a particular item. Once this prediction is generated, the system proceeds to offer recommendations or suggestions to the user based on the outcomes of these predictions.

Delving deeper into this mechanism reveals a complex interplay of algorithms and data analytics techniques aimed at deciphering user behaviour and preferences to deliver tailored recommendations.

1.3. How Does A Recommendation System Work?

At its core, the primary aim of a Recommendation System is to construct an objective function, or a mathematical model tailored to both end-users and specific items. Through this process, a Recommendation System is formed, and by iteratively optimizing the objective function, we can enhance the performance of the system.

The development of a Recommendation System typically involves three key steps: data loading and formatting, similarity computation between users or items, and prediction of unknown user ratings. Data collection may involve explicit or implicit means, with the system exhibiting improved optimization with an expansion in the dataset. Following data formatting, user-item interactions are commonly represented as a matrix known as a ratings matrix.

However, not all users have rated or expressed preference for every item, resulting in a sparse matrix. Various similarity measures are employed to infer missing ratings, and predictions of unknown ratings are made based on similarities between users or products. This process is repeated for each user in the dataset, with numerous methods available to compute these

unknown ratings.
In essence, these methodologies elucidate the fundamental workings of recommendation systems, underpinning their ability to generate personalized recommendations amidst vast datasets.

1.4. Types of Recommendation Systems?

The landscape of Recommendation Systems encompasses various methodologies tailored to suit diverse application requirements. These approaches can be broadly categorized into three main types: Collaborative Filtering, Content-Based Filtering, and Hybrid Recommendation Systems.

1.4.1. Collaborative Filtering Recommendation System

Collaborative Filtering Recommendation Systems harness user ratings to identify individuals with similar preferences and make predictions for unrated items based on the collective preferences of these like-minded users. This approach is further divided into User-based Collaborative Filtering (UBCF) and Item-based Collaborative Filtering (IBCF), focusing on similarities between users or items, respectively.
Implementation of Collaborative Filtering requires a rich dataset of user interactions and ratings, demonstrating versatility by not mandating detailed item information. However, it encounters challenges such as the cold-start problem for new users lacking sufficient rating information.

1.4.2. Content-Based Filtering Recommendation System

Content-Based Filtering Recommendation Systems generate content information for products, construct user profiles based on product features, and generate recommendations aligned with user preferences. This approach mitigates the cold-start problem by leveraging inherent item attributes to tailor recommendations.

1.4.3. Hybrid Recommendation System

Hybrid Recommendation Systems combine Collaborative Filtering and Content-Based Filtering strategies to enhance recommendation accuracy and coverage. These systems leverage innovative techniques such as weighted aggregation and feature combination to optimize recommendation efficacy, addressing multifaceted recommendation challenges.

2. Background

2.1. Collaborative Filtering Based Method

Collaborative filtering (CF) recommender systems are pivotal in contemporary recommendation frameworks, complementing Content-Based Filtering systems. They analyze user data to predict an individual's satisfaction with specific items, leveraging historical behaviors for insights. CF encompasses various approaches, including algorithms like Normal Predictor and Baseline Only, alongside k-Nearest Neighbors (k-NN) and Matrix Factorization techniques.
k-NN partitions data into clusters, recommending items based on similarity. Matrix Factorization, exemplified by Singular Value Decomposition (SVD), decomposes interaction matrices into smaller matrices, with ratings derived from their multiplication. Various training methods, including gradient descent and SVD-based approaches, minimize error between predicted and actual ratings.
To prevent overfitting, regularization factors are introduced, penalizing large vector magnitudes. Parameters like item and user biases are incorporated to refine predictions. Stochastic Gradient Descent (SGD) minimizes error iteratively, adjusting

parameters with a learning rate (γ). γ's choice balances convergence and computational efficiency.

2.1.1. User Based Collaborative Filtering (UBCF)

The user-based collaborative filtering (UBCF) approach acknowledges that utilizing all available user rating information might not be optimal. Therefore, the method strategically selects only the top-N similar users' information, enhancing the precision of model predictions. In UBCF, recommendations are formulated by leveraging the preferences within the user neighborhood, achieved through the following steps:

1. Identification of similar users based on shared preferences.
2. Provision of recommendations for new items to an active user, informed by ratings from similar users.

For instance, if a user, Ashley, rates both 'Star Wars' and 'The Empire Strikes Back' with five stars, and another user, Bob, also rates 'Star Wars' with five stars, it suggests a likelihood of similarity between Ashley and Bob. Consequently, the movie 'The Empire Strikes Back' could be recommended to Bob. Various methods quantify such similarities, as discussed in subsequent chapters.

2.1.2. Item Based Collaborative Filtering (IBCF)

In contrast to UBCF, item-based collaborative filtering (IBCF) generates recommendations relative to item neighborhoods. Initially, similarities between items are established, followed by recommending unrated items akin to those previously rated by the active user. The IBCF process unfolds in two stages:

1. Computation of item similarity based on their preferences.
2. Identification of the top similar items to the unrated items by the active user.

For example, upon determining a high similarity between 'Toy Story' and 'Aladdin,' recommending one movie to a user who appreciates the other effectively addresses the cold-start problem inherent in UBCF. This issue arises due to the system's inability to provide recommendations to first-time users lacking information within the system.

2.2. Content-Based Filtering Method

In content-based filtering, the content of an item serves as a pivotal determinant, offering a plethora of variables for consideration. For instance, in the context of a movie, variables such as genre, cast, director(s), and movie reviews can be factored into the algorithm. These variables can be utilized individually or in combination to enrich the algorithm's predictive capabilities.

Once the pertinent features are identified, the next step involves transforming this data into a Vector Space Model (VSM), an algebraic representation commonly used for text documents. This transformation is typically accomplished through a Bag of Words model, which disregards the word order within documents. In essence, each document is depicted as a "bag" containing a selection of words from a predefined dictionary.

The TF-IDF (Term Frequency-Inverse Document Frequency) representation serves as a specific implementation of the Bag of Words model. This model amalgamates the word's importance within the document (local importance) with its significance across the entire corpus (global importance). While historically entrenched in information retrieval systems, the concepts of TF and IDF are now gaining traction within content-based filtering recommenders. They aid in determining the relative importance of elements such as documents or movies.

An essential facet of content-based filtering is the utilization of similarity measures to gauge the likeness between items. Cosine Similarity stands out as a prominent measure in this regard, enabling the assessment of how closely items align with one another.

2.2.1. Similarity Measures

In the realm of recommendation systems, similarity measures play a crucial role in assessing the resemblance be-tween users or items. From a technical standpoint, these measures can be visualized on a plot, with each user or item rep-resented by coordinates. The distance between these coordinates serves as a metric for similarity, where shorter distances indicate greater likeness.

Cosine Similarity

Cosine similarity, a prevalent approach in recommendation systems, quantifies the similarity between two n-dimensional vectors by evaluating the angle between them in vector space. Applied to recommendation systems, these vectors represent items or users, with similarity assessed based on the angle between them. A smaller angle signifies higher similarity between items or users.

Consideration of a two-dimensional vector provides a foundational understanding. The dot product between two vectors equals the projection of one onto the other. Thus, for identical vectors, the dot product equals the square of their magni-tude, while for perpendicular vectors, the dot product is zero. In general, for n-dimensional vectors, the dot product can be computed as the summation of the product of corresponding components.

The dot product plays a pivotal role in defining similarity, directly influencing it. The similarity between two vectors u and v is determined by the ratio of their dot product to the product of their magnitudes. Cosine similarity, denoted as cos(θ), ranges between 0 and 1, indicating the extent of similarity between vectors. A value of 1 signifies identical vectors, while 0 indicates orthogonality.

$$similarity = \cos(\theta) = \frac{u\,.\,v}{||u||\,||v||} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\,\sqrt{\sum_{i=1}^{n} v_i^2}} \tag{1}$$

3. Evaluation Criterion

In order to assess whether a model is overfitting or underfitting, various evaluation techniques are employed. The primary objective of any model is to generalize well to future data. To achieve this, the dataset is typically divided into two subsets: training and test data. The training data is utilized to train the model, whereas the test data is reserved for evaluating its performance.

In an ideal scenario, the dataset is partitioned into a ratio of 80:20, with 80% allocated for training and 20% for testing. How-ever, fitting a linear model to non-linearly distributed data can result in underfitting, where the model performs poorly on both training and test data. Conversely, overfitting occurs when the model performs well on training data but poorly on test data, indicating that it fits excessively over the data distribution.

3.1. Cross Validation

Cross-validation is a statistical technique employed to estimate the performance of machine learning models. It encompasses two main types: exhaustive and non-exhaustive cross-validation. Exhaustive methods include leave-one-out and leave-p-out cross-validation, while non-exhaustive methods comprise k-fold cross-validation, Holdout method, and Repeated random sub-sampling validation. In this thesis, we focus on k-fold cross-validation.

This technique involves splitting the data into k groups, with each group serving as both training and testing data iteratively. The value of k denotes the number of groups the data is split into, hence referred to as k-fold cross-validation. It is widely used in machine learning to gauge a model's performance on unseen data.

Procedure:

1. Randomly shuffle the dataset.

2. Divide the dataset into k groups.

3. For each group:
- Use the group as test data.
- Utilize the remaining groups as training data.
- Train the model on the training set and evaluate it on the test set.
- Record the evaluation score and discard the model.

4. Summarize the model's performance using the collected evaluation scores.

This approach ensures that each observation in the dataset is allocated to a specific group, allowing it to be used in the test set once and in the training set k-1 times. Cross-validation provides a robust estimate of a model's generalization performance and is favored for its simplicity and unbiased estimation.

3.2. Root Mean Square Error (RMSE)

RMSE serves as a standard metric for assessing the accuracy of a model in predicting quantitative data. It is formally defined as:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{2}$$

Here, $\hat{y}1, \hat{y}2, \ldots, \hat{y}n$ are predicted values.

$y1, y2, \ldots, yn$ are observed values.

n is number of observations.

The division by n under the square root enables us to estimate the standard deviation σ of the error for a typical single observation, rather than measuring some form of "total error". By normalizing the error measure with respect to the number of observations, RMSE maintains consistency across different dataset sizes, becoming more accurate as the dataset size increases.

In essence, RMSE helps answer the question: "How much deviation should we expect in our model's next prediction?" RMSE is particularly useful for estimating the standard deviation σ of a typical observed value from the model's prediction, assuming that the observed data can be decomposed as:

$$\begin{aligned} &observed\ value \\ &\quad = predicted\ value \\ &\quad + predictably\ distributed\ random\ noise\ with\ mean\ zero \end{aligned} \tag{3}$$

The random noise represents any factors that the model fails to capture, such as unknown variables influencing the observed values. A small RMSE suggests that the model effectively predicts the observed data, while a large RMSE indicates that the model inadequately accounts for important underlying features in the data.

3.3. Mean Absolute Error (MAE)

MAE is one of the many metrics for summarizing and assessing the quality of a machine learning model. Here, error refers to the subtraction of Predicted value from Actual Value as below.

$$Prediction\ Error = Actual\ Value - Predicted\ Value \tag{4}$$

This prediction error is taking for each record after which we convert all error to positive. This is achieved by taking Absolute value for each error as below:

$$Absolute\ Error \rightarrow |Prediction\ Error| \tag{5}$$

Finally, we calculate the mean for all recorded absolute errors (Average sum of all absolute errors).

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{6}$$

Here, yi is the predicted value, xi is the actual value and n is the number of observations.

3.4. Qualitative and Quantitative Analysis

The comparison of various systems in this study encompasses two key dimensions. The quantitative aspect entails the utilization of metrics such as Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), as discussed in earlier sections. Conversely, the qualitative aspect centers on the quality of recommendations, assessed through visual inspection of the generated recommendations.

This dual approach allows for a comprehensive evaluation, combining numerical assessments with subjective judgments based on the perceived effectiveness and relevance of the recommendations.

4. Literature Survey

Recommender systems play a pivotal role in addressing the contemporary challenge of information overload by delivering personalized recommendations tailored to individual users. Over recent years, a plethora of approaches for constructing recommendation systems have emerged, leveraging Collaborative Filtering, Content-Based Filtering, or hybrid methodologies [4], [5], [6], [7].

Collaborative Filtering has attained a level of maturity and is widely adopted across diverse application domains. For instance, Group Lens, an architecture focused on news content, employs collaborative methods to assist users in navigating vast news databases, enhancing their article discovery experience [8]. Similarly, Amazon has refined its recommendation system by implementing topic diversification algorithms [9].

In contrast, Content-Based Filtering techniques concentrate on identifying similarities between content attributes and user preferences. These methods rely on user information to make predictions and disregard inputs from other users typical of collaborative techniques [10]. Notable examples include Letizia, which predicts user interest by tracking their interactions with websites, employing a Content-Based Filtering approach [11].

Despite the achievements of these filtering techniques, they are not without limitations. Content-Based Filtering methods encounter challenges such as restricted content analysis, overspecialization, and data sparsity [8], while collaborative approaches grapple with issues like cold-start problems, sparsity, and scalability. These challenges hinder their deployment in live production environments. In response, hybrid filtering has emerged as a solution, combining multiple filtering techniques in various configurations to enhance the performance and accuracy of recommender systems [12], [13]. Hybrid systems aim to capitalize on the strengths of each method while mitigating their inherent weaknesses [14].

5. Experiments

5.1. Dataset

In our analysis, we utilized the 'MovieLens 1M Dataset' [15], comprising 1,000,209 anonymous ratings assigned to approximately 3,900 movies by 6,040 MovieLens users who registered on the platform in 2000. We focused on two main files: ratings and movies. The ratings file consists of four fields: UserID, MovieID, Rating, and Timestamp.

- UserIDs span from 1 to 6040.
- MovieIDs span from 1 to 3952.
- Ratings adhere to a 5-star scale, considering only whole-star ratings.
- Timestamps are represented in seconds since the epoch.
- Each user has provided a minimum of 20 ratings.

The movies file comprises three fields: MovieID, Title, and Genres.

- Titles correspond to those listed on IMDB, including the release year.
- Genres are separated by pipes ('|') and are selected from a range of categories including Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, and Western.

We conducted preliminary exploratory analysis on the datasets. Figure 1 presents a histogram depicting the average ratings provided by users, showcasing a distribution approximating normality with a leftward skew. The majority of users' average ratings fall within the range of 3.5 to 4.

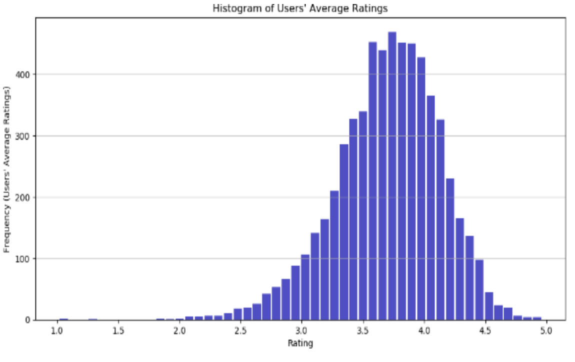

Figure 1. Histogram depicting the average ratings by users

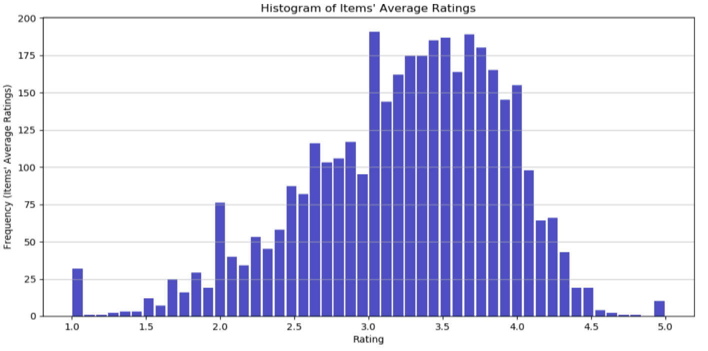

Figure 2. histogram displaying the average ratings received by items

Figure 2 illustrates the histogram displaying the average ratings received by items, also exhibiting a distribution resembling normality with a leftward skew. However, in this context, the ratings are more widely dispersed, with most items receiving ratings between 3 and 4.

Figure 3 showcases the histogram of ratings, maintaining consistency with the preceding plots by demonstrating that the most frequent ratings are 4 and 3, respectively.

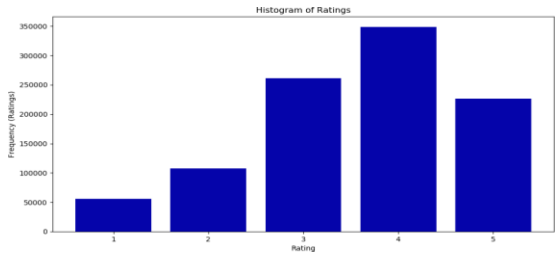

Figure 3. Histogram of Ratings

Figures 4 and 5 portray the histograms of items rated by users and users who rated items, respectively. As anticipated, these plots reveal that the majority of users rate only a small number of items.

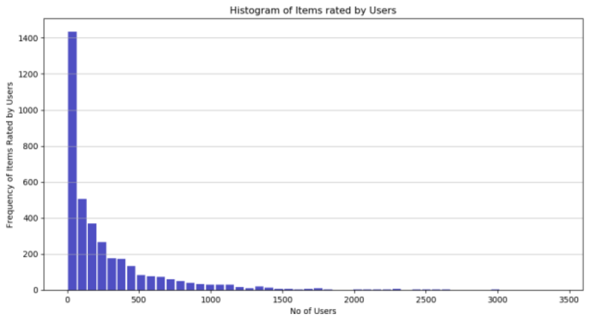

Figure 4. Items rated by users

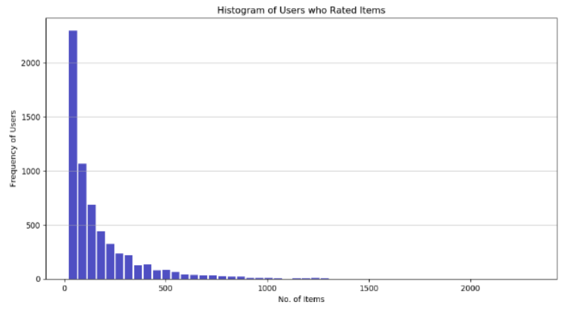

Figure 5. Users who rated items

Furthermore, we generated a word cloud depicting the genres of the movies. A word cloud offers a visual representation of textual data, wherein the prominence of each word is indicated by its font size or colour. This visualization aids in quickly identifying the most prevalent terms and assessing their relative importance. Figure 6 highlights some of the most popular genres, with drama and comedy emerging as the dominant categories.

Figure 6. Most Popular Genres

5.2. Methodology

5.2.1. Collaborative Filtering Based Recommendation

Singular Value Decomposition (SVD) paired with 5-fold cross-validation was utilized for Collaborative Filtering techniques. Given the recommendation system's aim to suggest movies to users, a method was devised to derive the top 20 recommendations for each user.

5.2.2. Content-Based Filtering Recommendation

For Content-Based Filtering, we utilized cosine similarity along with Term Frequency-Inverse Document Frequency (TF-IDF) applied to movie genres. Cosine similarity was employed to gauge the similarity between items, enabling us to derive the top 20 recommendations for individual users based on their preferences.

5.2.3. Hybrid Recommendation

The hybrid recommendation system takes a user ID and a movie name as input. Initially, Content-Based Filtering identifies movies most similar to the given one. Subsequently, Collaborative Filtering estimates ratings for these identified movies. The top-rated movies are then filtered and recommended to the user, amalgamating the strengths of both approaches to en-hance recommendation accuracy and relevance.

5.3. Result Analysis

5.3.1. Quantitative Analysis

We initiated our analysis by comparing the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) be-tween a Collaborative Filtering-based system and a Hybrid system. As Content-Based Filtering methods are primarily qualitative, we'll delve into them in the subsequent subsection.

We selected top-recommended movies from both systems for 10 users and computed the RMSE errors for each system for comparison. The RMSE plot for these users in Figure 7 indicates that the hybrid system demonstrates lower RMSE overall. Similarly, the average RMSE plot in Figure 8 further highlights the superiority of the hybrid system.

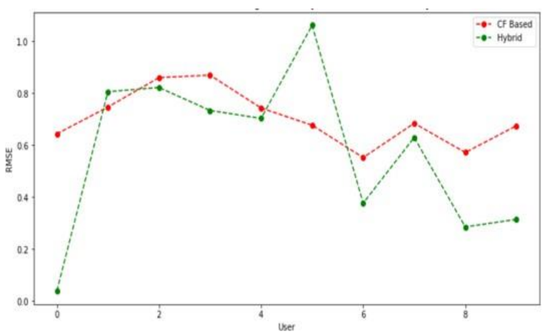

Figure 7. Indicates that the hybrid system demon-strates lower RMSE

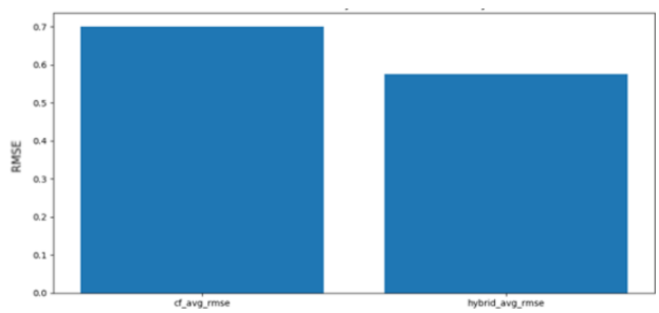

Figure 8. The average RMSE plot

Subsequently, we conducted the same evaluation for MAE. Figures 9 and 10 showcase that the hybrid recommendation sys-tem exhibits lower MAE, implying better accuracy.

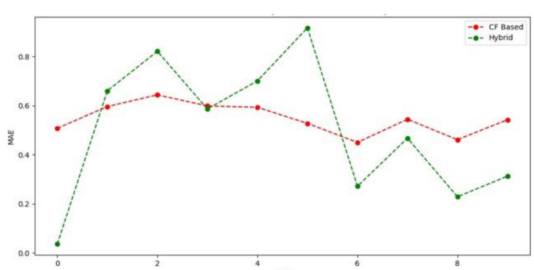

Figure 9. the hybrid recommendation system exhibits lower MAE, implying better accuracy

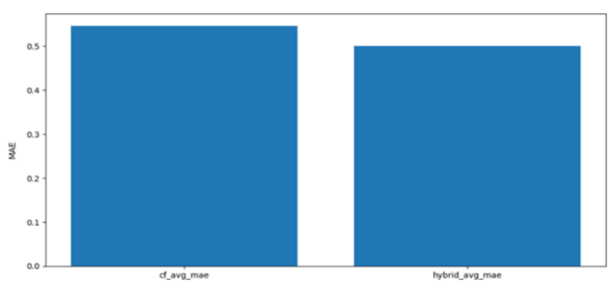

Figure 10. the hybrid recommendation system exhibits lower MAE, implying better accuracy

Further analysis involved evaluating 5 batches of users, each containing 5 users, to ascertain system performance. RMSE comparisons for these user sets, illustrated in Figure 11, demonstrate the hybrid system's comparative advantage. Correspondingly, Figure 12 portrays the average RMSE of Collaborative Filtering and Hybrid Recommendation System, affirming th evaluations for 5 user groups, depicted in Figures 13 and 14, consistently sh

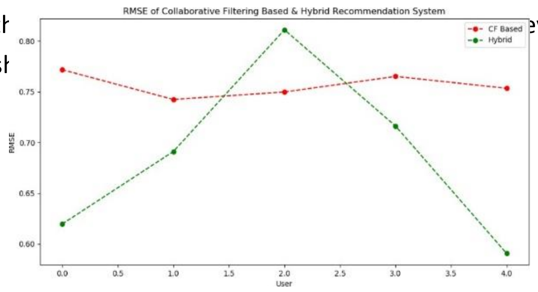

Figure 11. the hybrid system's comparative advantage

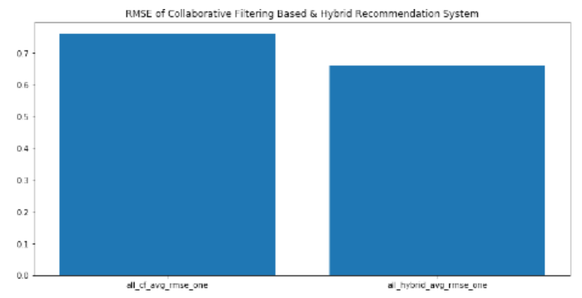

Figure 12. average RMSE of Collaborative Filtering and Hybrid Recommendation System

5.3.2. Qualitative Analysis

Qualitatively, Collaborative Filtering reveals movies a user is likely to rate highly but lacks the capability to recommend similar movies tailored to individual preferences. For instance, considering User 1, Table 2 demonstrates the top 20 recommended movies by the Collaborative Filtering system.

Table 2. Top 20 Recommended Movies for a Particular User by Collaborative Filtering Based Recommendation Systems

movie_id	estimated_rating	title	actual_rating	genres
527	4.995416	[Schindler's List (1993)]	[5]	[Drama|War]
318	4.958150	[Shawshank Redemption, The (1994)]	[]	[Drama]
1172	4.894460	[Cinema Paradiso (1988)]	[]	[Comedy|Drama|Romance]
2905	4.797475	[Sanjuro (1962)]	[]	[Action|Adventure]
1269	4.727696	[Arsenic and Old Lace (1944)]	[]	[Comedy|Mystery|Thriller]
920	4.695179	[Gone with the Wind (1939)]	[]	[Drama|Romance|War]
2019	4.693533	[Seven Samurai (The Magnificent Seven) (Shichi...	[]	[Action|Drama]
904	4.686336	[Rear Window (1954)]	[]	[Mystery|Thriller]
922	4.654332	[Sunset Blvd. (a.k.a. Sunset Boulevard) (1950)]	[]	[Film-Noir]
1234	4.651489	[Sting, The (1973)]	[]	[Comedy|Crime]
1203	4.645870	[12 Angry Men (1957)]	[]	[Drama]
905	4.644219	[It Happened One Night (1934)]	[]	[Comedy]
858	4.639405	[Godfather, The (1972)]	[]	[Action|Crime|Drama]
1197	4.635691	[Princess Bride, The (1987)]	[3]	[Action|Adventure|Comedy|Romance]
1233	4.632157	[Boat, The (Das Boot) (1981)]	[]	[Action|Drama|War]
356	4.629706	[Forrest Gump (1994)]	[]	[Comedy|Romance|War]
1198	4.626178	[Raiders of the Lost Ark (1981)]	[]	[Action|Adventure]
953	4.613348	[It's a Wonderful Life (1946)]	[]	[Drama]
912	4.610552	[Casablanca (1942)]	[]	[Drama|Romance|War]
1242	4.608209	[Glory (1989)]	[]	[Action|Drama|War]

Conversely, Content-Based Filtering recommends movies similar to a given one but doesn't predict whether a user will like them. As illustrated in Table 3, considering "Toy Story (1995)" as the reference movie, the top 20 recommended movies are shown. In conclusion, both qualitatively and quantitatively, a hybrid recommendation system outperforms standalone Collaborative or Content-Based Filtering systems.

Table 3. Top 20 Recommended Movies for a Particular Movie by Content-Based Filtering Recommendation System

movie_index	similarity_score	title	movie_id	genres
1050	1.000000	Aladdin and the King of Thieves (1996)	1064	[['Animation', "Children's", 'Comedy']]
2072	1.000000	American Tail, An (1986)	2141	[['Animation', "Children's", 'Comedy']]
2073	1.000000	American Tail: Fievel Goes West, An (1991)	2142	[['Animation', "Children's", 'Comedy']]
2285	1.000000	Rugrats Movie, The (1998)	2354	[['Animation', "Children's", 'Comedy']]
2286	1.000000	Bug's Life, A (1998)	2355	[['Animation', "Children's", 'Comedy']]
3045	1.000000	Toy Story 2 (1999)	3114	[['Animation', "Children's", 'Comedy']]
3542	1.000000	Saludos Amigos (1943)	3611	[['Animation', "Children's", 'Comedy']]
3682	1.000000	Chicken Run (2000)	3751	[['Animation', "Children's", 'Comedy']]
3685	1.000000	Adventures of Rocky and Bullwinkle, The (2000)	3754	[['Animation', "Children's", 'Comedy']]
236	0.869805	Goofy Movie, A (1995)	239	[['Animation', "Children's", 'Comedy', 'Romanc...
12	0.826811	Balto (1995)	13	[['Animation', "Children's"]]
241	0.826811	Gumby: The Movie (1995)	244	[['Animation', "Children's"]]
310	0.826811	Swan Princess, The (1994)	313	[['Animation', "Children's"]]
592	0.826811	Pinocchio (1940)	596	[['Animation', "Children's"]]
612	0.826811	Aristocats, The (1970)	616	[['Animation', "Children's"]]
700	0.826811	Oliver & Company (1988)	709	[['Animation', "Children's"]]
876	0.826811	Land Before Time III: The Time of the Great Gi...	888	[['Animation', "Children's"]]
1010	0.826811	Winnie the Pooh and the Blustery Day (1968)	1023	[['Animation', "Children's"]]
1012	0.826811	Sword in the Stone, The (1963)	1025	[['Animation', "Children's"]]
1020	0.826811	Fox and the Hound, The (1981)	1033	[['Animation', "Children's"]]

Table 4. Top 20 Recommended Movies for a Particular User and Movie by Hybrid Recommendation

movie_index	similarity_score	title	estimated_rating	actual_rating
2073	1.000000	American Tail: Fievel Goes West, An (1991)	4.107195	[]
1050	1.000000	Aladdin and the King of Thieves (1996)	4.077366	[]
2285	1.000000	Rugrats Movie, The (1998)	4.061477	[]
3685	1.000000	Adventures of Rocky and Bullwinkle, The (2000)	4.052081	[]
3542	1.000000	Saludos Amigos (1943)	3.732172	[]
3682	1.000000	Chicken Run (2000)	3.595624	[]
3045	1.000000	Toy Story 2 (1999)	3.319540	[]
2072	1.000000	American Tail, An (1986)	3.047335	[]
2286	1.000000	Bug's Life, A (1998)	2.749218	[]
236	0.869805	Goofy Movie, A (1995)	3.779034	[]
1949	0.826811	Bambi (1942)	4.424280	[]
2731	0.826811	Little Nemo: Adventures in Slumberland (1992)	4.384338	[]
2618	0.826811	Tarzan (1999)	4.315832	[]
2070	0.826811	Secret of NIMH, The (1982)	4.288048	[]
3730	0.826811	Pokémon the Movie 2000 (2000)	4.255108	[]
1012	0.826811	Sword in the Stone, The (1963)	4.199251	[]
2068	0.826811	Charlotte's Web (1973)	4.140761	[]
2692	0.826811	Iron Giant, The (1999)	4.049191	[4]
592	0.826811	Pinocchio (1940)	3.935920	[]
3546	0.826811	Dinosaur (2000)	3.922755	[]

6. Conclusion and Future Scope

In this study, I explored various recommendation systems, including Collaborative Filtering, Content-Based Filtering, and a Hybrid recommendation system, using the well-known MovieLens dataset. By conducting a qualitative and quantitative analysis, I compared these three recommendation systems. This mixed analysis approach was necessary as Content-Based Filtering systems are not easily quantifiable, and for a movie recommendation system, the qualitative aspect holds significant importance. Therefore, I devised a method that combined both quantitative and qualitative evaluations.

Through our analysis, it became evident that a hybrid recommendation system consistently outperforms standalone methods in terms of recommendation accuracy and relevance.

There are several avenues for further research building upon this study. For instance, we did not incorporate demographic information about users into our recommendation system. Integrating such data could enhance the accuracy of the hybrid recommendation system by providing additional insights into user preferences. Additionally, while we considered only movie genres in Content-Based Filtering recommendations, exploring other factors such as cast, crew, and reviews could further re-fine similarity assessments.

Furthermore, conducting a comparative analysis among different Collaborative Filtering methods and similarity measures could yield valuable insights into their effectiveness and applicability in different contexts. Overall, the study opens up numerous possibilities for future research to enhance the performance and capabilities of recommendation systems.

References

[1]. V. M, and T. K, “History and overview of the recommender systems”, in Collaborative Filtering Using Data Mining and Analysis, V. Bhatnagar, Ed. Hershey: *IGI Global*, pp. 74-99, 2016.

[2]. M. Rouse, “What is collaborative filtering?”, WhatIs.com, para. 1, Aug. 11, 2017, [Online], Available: https://whatis.techtarget.com/definition/collaborative-filtering

[3]. C.S. Perone, “Machine Learning :: Cosine Similarity for Vector Space Models (Part III)”, Dec. 09, 2013, [Online], Available: http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector- space-models-part-iii/

[4]. B. Sarwar, G. Karypis, J. Konstan and R. John, “Item-based Collaborative Filtering recommendation algorithms”, 2001, pp. 285-295.

[5]. Z. Zhao, M. Shang, “User-based collaborative-filtering recommendation algorithms on Hadoop”, in *Third International Conference on Knowledge Discovery and Data Mining, Phuket, Thailand*, Jan. 09, 2010, IEEE, pp. 478-481.

[6]. R. V. Meteren, M. V. Someren, “Using Content-Based Filtering for recommendation”, 2000.

[7]. Y. Shih and D. Liu, “Hybrid recommendation approaches: Collaborative Filtering via valuable content information”, in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences, Big Island, USA, Jan. 06, 2005, IEEE*, pp. 217b-217b.

[8]. G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions”, in *IEEE Transactions on Knowledge & Data Engineerin*g, pp. 734-749, Apr. 25, 2005.

[9]. C. Ziegler, S. M. McNee, J. A. Konstan and G. Lausen, “Improving recommendation lists through topic diversification”, in *Proceedings of the 14th international conference on World Wide Web, Chiba, Japan*, *ACM*, pp. 22-32, May. 14, 2005.

[10]. S. Min and I. Han, “Detection of the customer time-variant pattern for improving recommender systems”, *Expert systems with applications*, vol. 28, no. 2, pp. 189-199., Nov. 2004

[11]. H. Lieberman, “Letizia: An Agent That Assists Web Browsing”, *IJCAI*, pp. 924-929, 1995.

[12]. B. Mobasher, “Recommender Systems,” *Kunstliche Intelligenz*, Special Issue on Web Mining, *BottcherIT Verlag, Bremen, Germany*, pp. 41-43, 2007.

[13]. M. Göksedef and S. G. Öğüdücü, “Combination of Web page recommender systems”, *Expert systems with applications*, vol. 37, no. 4, pp. 2911-2922., Apr 2010.

[14]. M. Y. H. Al-Shamri and K. K. Bharadwaj, “Fuzzy-genetic approach to recommender systems based on a novel hybrid user model”, *Expert systems with applications*, vol. 35, no. 3, pp.1386-1399., Oct. 2008.

[15]. F. M. Harper and J.A. Konstan, “The MovieLens Datasets:History and Context”, ACM Transactions on Interactive Intelligent Systems (TiiS) - *Regular Articles and Special issue on New Directions in Eye Gaze for Interactive Intelligent Systems (Part 1 of 2)*, Article 19, vol. 5, no. 4, pp. 12-14., Jan. 2016

[16]. P. Singh, G. Srivastava, S, Singh, & S. Kumar, “Intelligent movie recommender framework based on content-based & collaborative filtering assisted with sentiment analysis”, I*nternational Journal of Advanced Research in Computer Science*, vol. 14 no. 3, 2023.

[17]. B. Hazela, P. Asthana, S. Singh, and G. Srivastava, “Enhance Movie Recommender System Using Machine Learning Techniques”, *Proceedings of the Advancement in Electronics & Communication Engineering*, 2022, Available at SSRN: https://ssrn.com/abstract=4159215 or http://dx.doi.org/10.2139/ssrn.4159215

[18]. G. Srivastava, V. Singh, S. Kumar, “Opinion Mining-Assisted Intelligent Program Selection Employing Fuzzy SWARA Mechanism”. In: Tanwar, S., Wierzchon, S.T., Singh, P.K., Ganzha, M., Epiphaniou, G. (eds) *Proceedings of Fourth International Conference on Computing, Communications, and Cyber-Security. CCCS 2022. Lecture Notes in Networks and System*s, vol 664, 2023. Springer, Singapore. https://doi.org/10.1007/978-981-99-1479-1_45.