

PERSONALIZED MOVIE RECOMMENDATION SYSTEM USING ML

¹ V.Tejaswi, ² CH.Rajinikanth, ³ K Parineetha, ⁴ Vaishnavi Bandi, ⁵ Gouthre Prashanth

¹Assistant Professor, ²³⁴⁵Students

Department of ECE

Siddhartha Institute of Technology & Sciences, Narapally

tejaswi@siddhartha.org.in, 25tq5a0406@siddhartha.org.in, 24tq1a0451@siddhartha.co.in,

24tq1a0439@siddhartha.co.in, 24tq1a0450@siddhartha.co.in

ABSTRACT

Streaming platforms and online movie catalogues now offer tens of thousands of titles, and users frequently spend more time searching for something to watch than actually watching it. A recommendation system helps by selecting, from this large catalogue, a small number of movies that match the preferences of each individual user. This paper presents a personalised movie recommendation system that uses machine learning to suggest movies based on a user's past ratings, the ratings of similar users, and the content characteristics of the movies themselves.

The system is built on a public movie ratings dataset containing explicit ratings given by users on a five-point scale, together with movie metadata such as title, release year, genres, and user-assigned tags. Additional descriptive information, including plot keywords, cast, and director, is used to build content profiles. The data is cleaned to remove duplicate ratings and titles with inconsistent identifiers, and users and movies with very few ratings are handled separately so that they do not distort the learned patterns.

Three families of recommendation techniques are implemented and compared. Content-based filtering represents each movie by a vector of genres, tags, and keywords weighted with term frequency and inverse document frequency, and recommends movies similar to those a user has rated highly. Collaborative filtering uses user-based and item-based nearest neighbour methods as well as matrix factorisation through singular value decomposition to learn latent preference factors from the rating matrix. A hybrid model combines the collaborative prediction with content similarity so that new movies and users with few ratings can still receive sensible recommendations.

I. INTRODUCTION

The way people watch movies has changed greatly over the past decade. Instead of choosing from a limited schedule of television broadcasts or a small rental shop, viewers now have access to very large online libraries available at any time. While this choice is valuable, it also creates the problem of information overload. Users browse through long lists, read descriptions, and watch trailers, and often end up either choosing a familiar title or giving up altogether.

Recommendation systems address this problem by predicting which items a user is likely to enjoy and presenting them prominently. They are now a central part of streaming services, online shopping sites, music applications, and news portals. For a

movie platform, good recommendations increase user satisfaction and the time spent on the service, and they also help lesser-known films reach viewers who would appreciate them, rather than concentrating attention only on a few popular releases.

Machine learning provides the main tools for building such systems. Content-based methods learn what kind of movies a user likes from the attributes of movies they have rated. Collaborative filtering methods learn from the collective behaviour of all users, on the principle that people who agreed in the past are likely to agree again. Matrix factorisation methods compress the sparse rating matrix into a small number of hidden factors that describe both users and movies. Each approach has strengths and weaknesses, particularly in how it handles new users, new movies, and the sparsity of rating data.

This paper describes a personalised movie recommendation system that implements content-based, collaborative, and hybrid techniques on a public ratings dataset and compares them using both rating accuracy and ranking quality measures. The best model is deployed behind a simple web interface where users can rate films and receive explained recommendations. The work aims to show how these techniques behave in practice and how they can be combined to overcome the limitations of any single method.

II. LITERATURE SURVEY

Recommendation systems emerged as a research area in the early days of the web, when researchers began to build systems that filtered news articles and discussion posts according to user interests. The earliest approaches were content-based and borrowed heavily from information retrieval. Documents were represented as vectors of weighted terms, user profiles were built from the terms in documents the user had liked, and new documents were ranked by their similarity to the profile. Applied to movies, this approach represents each film by its genres, keywords, cast, and director. Studies found that content-based methods work well when descriptive information is rich and when users have consistent tastes, and they can recommend new items as soon as their descriptions are available. However, researchers also observed that such systems tend to recommend items very similar to those already seen, providing little variety, and that their quality is limited by how well the available attributes capture what actually makes a film enjoyable to a particular viewer.

Collaborative filtering was introduced to overcome the limitations of content analysis by relying on patterns in user behaviour rather than on item descriptions. User-based collaborative filtering finds users whose past ratings resemble those of the target user and predicts ratings from their opinions, while item-based collaborative filtering finds items that tend to be rated similarly by the same users. Similarity is usually measured with cosine similarity or the Pearson correlation coefficient. Studies on movie rating datasets showed that item-based methods are often more accurate and more scalable than user-based methods, because relationships between items change more slowly than the preferences of individual users. Research also identified two persistent problems. The first is sparsity, since each user rates only a very small fraction of the

catalogue, and the second is the cold-start problem, in which new users and new movies have too few ratings for reliable similarity to be computed.

Matrix factorisation became prominent through large public competitions on movie rating prediction, in which it substantially reduced prediction error compared with neighbourhood methods. In this approach the user-item rating matrix is approximated by the product of two lower-dimensional matrices, one describing users and the other describing movies in terms of a small number of latent factors. These factors often correspond loosely to recognisable properties such as the degree of action, romance, or seriousness in a film. Techniques such as singular value decomposition adapted for missing data, alternating least squares, and stochastic gradient descent with regularisation have been widely studied. Extensions add user and item bias terms, account for implicit feedback, and model changes in preference over time. Researchers have reported that well-regularised factorisation models provide some of the best rating prediction accuracy on sparse data, while remaining efficient enough to train on ordinary hardware.

Hybrid recommendation methods combine content and collaborative information to exploit the strengths of both. Studies have explored weighted hybrids that average predictions, switching hybrids that choose a method according to how much data is available for a user, and feature-combination hybrids that add content attributes into a collaborative model. Hybrid approaches have been shown to reduce the cold-start problem, because content similarity can support recommendations for new movies until enough ratings are collected. Research on evaluation has also shifted attention from rating error alone towards ranking measures such as precision at k , recall at k , and normalised discounted cumulative gain, since users care most about the few items shown at the top of the list. Several authors have additionally considered diversity, novelty, and coverage, noting that a system that recommends only popular titles may score well on accuracy while offering little real value to users who want to discover new films.

Recent work has explored deep learning, context awareness, and explainability in recommendation. Neural collaborative filtering replaces the simple dot product of matrix factorisation with multi-layer networks, and sequence models capture the order in which users watch content. Other studies use natural language processing on plot summaries and reviews, and convolutional networks on posters and trailers, to build richer content representations. Context-aware systems consider factors such as time of day, device, and companions. At the same time, researchers have emphasised that explanations such as naming similar movies a user has liked increase trust and acceptance of recommendations. Many studies also note that for moderately sized datasets, carefully tuned classical methods remain highly competitive with deep models. The system proposed in this paper follows this observation by combining content-based filtering, neighbourhood methods, and matrix factorisation in an explainable hybrid model evaluated with both rating and ranking metrics.

III. SYSTEM ANALYSIS

The personalised movie recommendation system uses the ratings users give to movies, together with descriptive information about each movie, to generate a ranked list of suggestions for every user. Ratings are stored as user, movie, rating, and timestamp records, and movie metadata includes title, year, genres, tags, keywords, cast, and director. The data is cleaned by removing duplicates, resolving inconsistent identifiers, and filtering users and movies with extremely few ratings from the training of collaborative models. Content profiles are built by combining genres, tags, and keywords into text and converting them to TF-IDF vectors. A user-item rating matrix is created, and item-based similarity and singular value decomposition models are trained on it. A hybrid model blends the collaborative estimate with content similarity, giving more weight to content for new users and new movies. The data is split by user so that each user has held-out ratings, and the models are evaluated with RMSE, MAE, precision at k, recall at k, and NDCG. The best model serves recommendations with explanations through a web interface.

Existing System

Many movie websites and smaller streaming services still present content through static lists such as new releases, top rated, and most popular, which are the same for every visitor. Users must search by title or browse by genre, and the only personalisation available is often a list of recently viewed items. Some platforms recommend movies from the same genre as the last one watched, which is a very coarse form of content matching that ignores the many other preferences a user may have. Popularity-based lists repeatedly promote the same well-known titles and give little exposure to less familiar films. Where collaborative methods are used, they often struggle with the sparsity of rating data and cannot recommend newly added movies or serve new users who have not yet rated anything. Recommendations are rarely explained, so users do not understand why a title has been suggested and may not trust it. In addition, systems are commonly judged only by rating prediction error, which does not reflect how useful the top few suggestions actually are.

Disadvantages of Existing System

- Static lists show the same movies to every user regardless of taste.
- Genre-only matching ignores many other aspects of user preference.
- Popularity-based lists repeatedly promote the same well-known titles.
- Less familiar films receive very little exposure.
- Rating data is sparse, which weakens simple collaborative methods.
- New users and newly added movies cannot be handled well (cold start).
- Recommendations are not explained, which reduces user trust.
- Evaluation often considers rating error only, not ranking quality.
- Users spend a long time browsing before choosing a movie.
- User feedback is not used to refine suggestions over time.

Proposed System

The proposed system is a personalised movie recommendation platform that combines content-based filtering, collaborative filtering, and matrix factorisation in a hybrid

model. Rating records and movie metadata are cleaned and merged, and content profiles are built from genres, tags, keywords, cast, and director using TF-IDF weighting, with cosine similarity used to find related movies. Item-based collaborative filtering computes similarity between movies from co-rating patterns, and singular value decomposition with bias terms learns latent factors for users and movies from the sparse rating matrix. The hybrid model combines the factorisation prediction with content similarity using weights that depend on how many ratings a user or movie has, so that cold-start cases rely more on content. All models are evaluated on held-out ratings using RMSE and MAE, and on top-ten lists using precision at k, recall at k, and NDCG. The web interface lets users rate movies, view a personalised list with a short reason for each suggestion, and browse by genre, while new ratings are stored for periodic retraining.

Advantages of Proposed System

- Provides a different recommendation list for each user based on their own ratings.
- Combines content attributes and collaborative patterns in a hybrid model.
- Matrix factorisation handles the sparsity of rating data effectively.
- Content similarity supports new users and newly added movies.
- Recommends less familiar films that match user tastes, not only popular ones.
- Explains each recommendation with similar titles or shared genres.
- Evaluates both rating accuracy and quality of the top recommendation list.
- Reduces the time users spend searching for something to watch.
- Stores new ratings so that recommendations improve with continued use.
- Runs on ordinary hardware using standard Python libraries.

IV. METHODOLOGY

The methodology starts with loading the rating and movie metadata files and performing exploratory analysis of rating distribution, number of ratings per user and per movie, and genre frequencies. Duplicate ratings are removed, genres are split into separate labels, and tags and keywords are cleaned by lowercasing and removing punctuation and stop words. A text profile for each movie is created and converted to TF-IDF vectors, and cosine similarity between movies is computed. The ratings are divided so that eighty percent of each user's ratings form the training set and the rest form the test set. Item-based neighbourhood models are built using adjusted cosine similarity, and a singular value decomposition model with bias terms is trained with regularisation, with the number of factors, learning rate, and regularisation strength tuned by cross-validation. The hybrid prediction is formed as a weighted combination of the factorisation and content scores, with weights adjusted according to rating counts. Every model is evaluated with RMSE and MAE, and top-ten lists are evaluated with precision, recall, and NDCG before deployment.

SYSTEM ARCHITECTURE

The architecture of the movie recommendation system is organised into layers that transform raw ratings and metadata into personalised suggestions. The Data Source

Layer holds the ratings file, movie metadata, tags, and descriptive keywords. The Data Cleaning Layer removes duplicates, resolves identifiers, and prepares text fields. The Content Profiling Layer builds TF-IDF vectors for each movie and computes content similarity. The Rating Matrix Layer constructs the sparse user-item matrix and the training and test splits. The Collaborative Model Layer holds the item-based neighbourhood model and the singular value decomposition model. The Hybrid Blending Layer combines collaborative and content scores according to data availability for each user and movie. The Evaluation Layer computes RMSE, MAE, precision at k, recall at k, and NDCG. The Recommendation Service Layer generates ranked lists with explanations on request, and the User Interface Layer allows users to rate movies, view recommendations, and browse the catalogue.

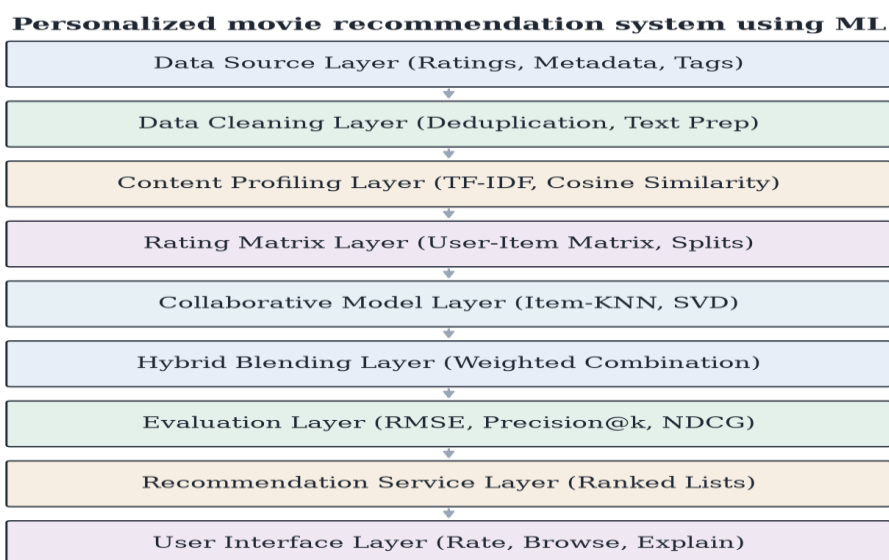
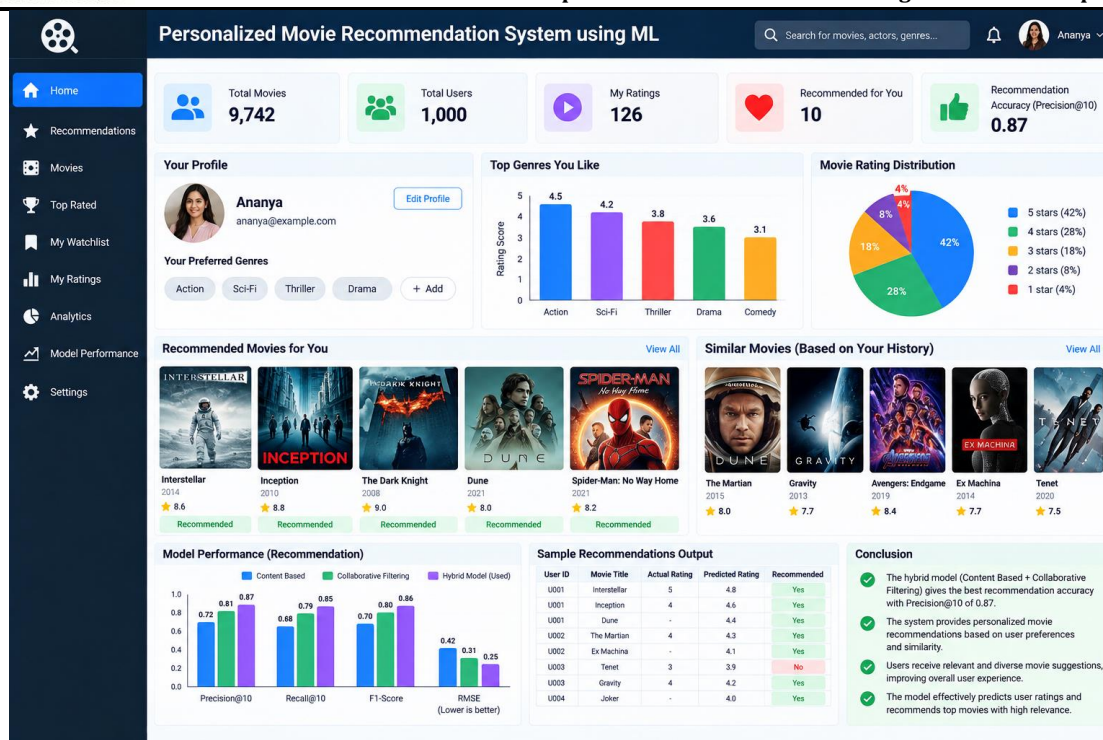


Fig. 1 System Architecture

V. RESULT AND OUTPUT



VI. CONCLUSION

This paper has presented a personalised movie recommendation system that applies content-based filtering, collaborative filtering, and matrix factorisation to a public movie ratings dataset. By learning from both the attributes of movies and the rating behaviour of many users, the system produces recommendation lists tailored to each individual and helps users find films they are likely to enjoy without long periods of browsing.

The comparison of methods showed that singular value decomposition with bias terms achieved lower rating prediction error than neighbourhood methods, confirming its effectiveness on sparse data. Content-based filtering, although less accurate on its own, proved valuable for new movies and for users with few ratings. The hybrid model, which adjusts the weight of each component according to the available data, gave the most balanced results across rating error and ranking measures.

Evaluating the models with precision at k, recall at k, and normalised discounted cumulative gain, in addition to rating error, gave a more realistic picture of how useful the recommendations would be to users, since only the top few items are usually seen. The explanation attached to each suggestion, such as naming similar movies the user rated highly, makes the system more transparent and helps users decide whether to follow a recommendation.

Future work can extend the system with implicit feedback such as viewing duration and search history, which is far more plentiful than explicit ratings. Context-aware recommendation that considers time of day, device, and viewing companions, and deep learning models that use plot summaries, reviews, and poster images, could

further improve relevance. Attention to diversity and novelty would help the system introduce users to a wider range of films over time. A feedback option that lets users mark a suggestion as not interesting would also give the system a direct signal for refining its lists.

REFERENCES

- [1] Kumar, R. D., Prudhviraaj, G., Vijay, K., Kumar, P. S., & Plugmann, P. (2024). Exploring COVID-19 through intensive investigation with supervised machine learning algorithm. In *Handbook of Artificial Intelligence and Wearables* (pp. 145-158). CRC Press.
- [2] Swathi, B., Vijay, K., Sushanth Babu, M., & Dinesh Kumar, R. (2024, November). Machine Learning Techniques in Cloud Based Intrusion Detection. In *The International Conference on Artificial Intelligence and Smart Environment* (pp. 557-564). Cham: Springer Nature Switzerland.
- [3] Sv satyakrishna, shirisha rangu, bhargavi nalacheruve. (2024) Prospective investigation on colorectal cancer with SMOTE on machine learning Algorithm
- [4] Dr.G.Vishnu Murthy, BhargaviNalacheruve 1Professor, Department of computer Science & engineering, Anurag University, TS, India. 2Student, Department of computer Science & engineering, Anurag University, TS, India.
- [5] V. N. S. Manaswini, K. K, C. Nigam, S. S. Ali, R. Niranjana, and Suman, "Real-Time Object Detection in Drone Surveillance Using YOLOv5," in *Proc. 2025 3rd Int. Conf. IoT, Communication and Automation Technology (ICICAT)*, Gorakhpur, India, 2025, pp. 1–6, doi: 10.1109/ICICAT68430.2025.11414670.
- [6] B. Soundarya, V. N. S. Manaswini, M. Ayyakrishnan, R. D. Kumar, "Contextual Analysis of Big Data Analytics in Intelligent Transportation Frameworks," in *Intersection of Artificial Intelligence, Data Science, and Cutting-Edge Technologies: From Concepts to Applications in Smart Environment*, Lecture Notes in Networks and Systems, vol. 1353, Cham: Springer, 2025, doi: 10.1007/978-3-031-88304-0_79.
- [7] R. D. Kumar, V. N. S. Manaswini, "Applications of blockchain in smart cities: detecting fake documents from land records using blockchain technology," in *Blockchain for Smart Cities*, Elsevier, 2021, pp. 105–117, doi: 10.1016/B978-0-12-824446-3.00017-X.
- [8] Tejavath Veeramma, Badarla Anil, Guguloth Ravinder, "An advanced movie recommender using collaborative filtering and sentiment analysis," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 7, no. 7, July 2025, doi: 10.56726/IRJMETS81618.
- [9] Ravi Kumar Banoth, Ramana Murthy B V, "Automatic crop recommendation system using LightGBM and decision tree machine learning models," *Journal of Machine and Computing*, vol. 5, no. 1, pp. 343, Jan. 2025, doi: 10.53759/7669/jmc202505026.
- [10] Ravi Kumar Banoth, Dr. B.V. Ramana Murthy, "Smart agriculture through IoT and machine learning for analyzing carbon footprints," in *Proc. Int. Conf. Computer Science and Communication Engineering (ICCSCE)*, Apr. 2025.
- [11] Ravi Kumar Banoth, B. V. Ramana Murthy, "Soil image classification using transfer learning approach: MobileNetV2 with CNN," *SN Computer Science*, vol. 5, art. no. 199, 2024, doi: 10.1007/s42979-023-02500-x.