Movie Recommendation System

Maulik Desai[1] Abhinav Gupta[1] Tejas Gupta[1] Harshit Yadav[1]
Samay Mehar[1] Mohit Kumar[1]

[1]Indian Institute of Technology, Jodhpur

Abstract

In the contemporary landscape of digital media consumption, the vast array of available movies presents a challenge for users seeking personalized recommendations. In response, this project introduces a machine learning-based movie recommendation system designed to address this challenge. Our system leverages user preferences and historical movie ratings to offer tailored recommendations, thereby enhancing user engagement and satisfaction with movie platforms. We employ collaborative filtering, content-based filtering, and matrix factorization techniques to generate accurate and diverse movie suggestions. Additionally, we conduct thorough performance evaluations using various metrics and compare our system with existing recommendation approaches. The results demonstrate the efficacy and efficiency of our method in delivering relevant movie recommendations, with implications for enhancing user satisfaction and engagement in movie consumption platforms.

Keywords: movie recommendation, machine learning

Contents

1 Introduction

In today's digital age, the realm of entertainment has expanded exponentially, particularly with the proliferation of streaming platforms offering vast libraries of movies. While this abundance provides viewers with countless options, it also presents a challenge: how does one navigate through this sea of content to find movies that match their interests and preferences?

This challenge is exacerbated by the sheer diversity of movie genres, styles, and themes available, making it increasingly difficult for users to discover new films they might enjoy. Traditional methods of browsing through categories or relying on generic recommendations often fall short in providing truly personalized suggestions.

To address this challenge, we present a comprehensive exploration and implementation of a Movie Recommendation System. Leveraging the power of machine learning and matrix factorization [5] techniques from sci-kit learn [4], our system aims to provide users with personalized movie recommendations tailored to their tastes. The objective is to enhance user experience, increase engagement, and ultimately, improve user satisfaction with movie consumption platforms.

Our recommendation system employs a combination of collaborative filtering and content-based filtering techniques [2] to analyze user preferences and historical movie ratings. Collaborative filtering relies on the idea that users who have liked similar movies in the past will likely enjoy similar movies in the future. Content-based filtering, on the other hand, recommends movies based on the features of the items and the user's preferences.

This report delves into various aspects of our Movie Recommendation System. We provide an overview of the methodologies employed, discussing the advantages and limitations of each approach. We also detail the dataset used for training and testing our system, as well as the preprocessing steps involved.

Furthermore, we conducted a series of experiments to evaluate the performance of our recommendation system. These experiments involved comparing different algorithms, tuning parameters, and assessing the accuracy and diversity of recommendations. The results obtained provide valuable insights into the effectiveness of our approach and its potential for real-world applications.

In addition to discussing the technical aspects of our system, we also explore the implications of our findings. We analyze the impact of personalized recommendations on user engagement and satisfaction, as well as the potential for enhancing business metrics such as user retention and revenue generation for movie platforms.

Finally, we summarize our findings and outline potential areas for further research. By contributing to the ongoing discourse on recommendation systems, particularly in the realm of movie recommendations, we aim to provide valuable insights for researchers and practitioners alike.

2 Approaches Tried

There are majorly two broad ways that we explored for movie recommendation. One is content based filtering, and the other is collaborative filtering.

- **Collaborative filtering:** Relies on the idea that users who have liked similar movies in the past will likely enjoy similar movies in the future.
- **Content-based filtering:** Recommends movies based on the features of the items and the user's preferences.

Let us consider *Table 1* which studies the ratings given by different users to different movies, to help us understand collaborative and content based filtering better.

Table 1: User Ratings for Movies

User	**Movie A**	**Movie B**	**Movie C**	**Movie D**
User A	**2**	**5**		**4**
User B		**2**		**3**
User C	**3**	**5**	**1**	**4**

From the ratings of movies A, B and D we can see that the user A and user C seem to give similar ratings to movies, so we can conclude to an extent that the taste of user A is similar to the taste of user C in movies. This deduction of similarity in tastes of users is **Collaborative Filtering** So if we consider movie C, we see that user C has rated the movie poorly, we can say that user A is also likely to dislike movie C.

If we get to know that movie B is similar to movie C then **Content-based Filtering** tells us that user A will like movie C as well, because they liked movie B.

2.1 Collaborative Filtering

For collaborative filtering, we used two major approaches, first is traditional machine learning models, the other is Surprise [3].

2.1.1 Traditional ML Models

We used the following ML models:-

- **K Nearest Regressor:** KNN regression is a non-parametric method that, in an intuitive manner, approximates the association between independent variables and the continuous outcome by averaging the observations in the same neighbourhood.
- **Support Vector Regressor:** Support vector regression (SVR) is a type of support vector machine (SVM) that is used for regression tasks. It tries to find a function that best predicts the continuous output value for a given input value.
- **Decision Tree Regressor:** Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs, and utility.

- **Singular Value Decomposition:** The Singular Value Decomposition (SVD) of a matrix is a factorization of that matrix into three matrices.
- **Random Forest Regressor:** Random Forest Regression is a versatile machine-learning technique for predicting numerical values. It combines the predictions of multiple decision trees to reduce overfitting and improve accuracy.
- **Gradient Boosting Regressor:** Gradient Boosting is a powerful boosting algorithm that combines several weak learners into strong learners, in which each new model is trained to minimize the loss function such as mean squared error of the previous model using gradient descent.
- **Bagging Regressor:** Bagging (or Bootstrap aggregating) is a type of ensemble learning in which multiple base models are trained independently and in parallel on different subsets of the training data. Each subset is generated using bootstrap sampling, in which data points are picked at random with replacement.

2.1.2 Surprise

Surprise is a Python scikit for building and analyzing recommender systems that deal with explicit rating data. It gives users perfect control over their experiments. To this end, a strong emphasis is laid on documentation, which they have makde as clear and precise as possible by pointing out every detail of the algorithms. We used this library to help enhance our movie recommendation system. The Surprise library contains multiple implementations of various machine learning models of its own and we used many of them to figure out and select the best one for our use.

2.2 Content-based Filtering

For content-based filtering we use cosine similarity. Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis.

Tf-idf, short for term frequency–inverse document frequency [1], is a measure of importance of a word to a document in a collection or corpus, adjusted for the fact that some words appear more frequently in general. We used sci-kit learn's TFidfVectorizer to get vectors of the various genres and used these vectors and their cosine similarities to get content based predictions, such that similar movies get recommended.

3 Experiments and Results

We used the MovieLens dataset for this project. The MovieLens dataset is a well-known benchmark dataset widely used in the field of recommendation systems. It contains movie ratings provided by users of the MovieLens website, along with movie metadata. The dataset is frequently used for research and evaluation of recommendation algorithms due to its size, diversity, and availability.

3.1 Data Visualization

We utilized various data visualization techniques to gain insights into the MovieLens dataset. Here's a brief summary of the visualizations:

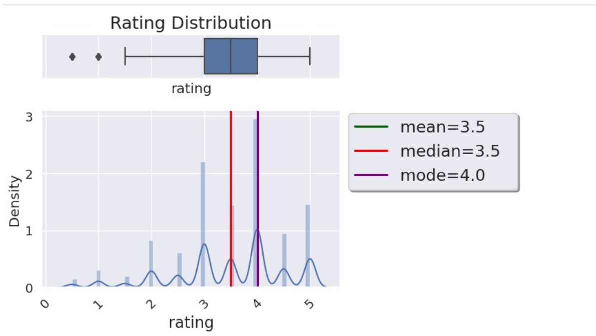

Figure 1: Distribution of Rating Variable

This graph shows the rating distribution of the dataset. The x-axis represents the rating values, and the y-axis shows the density of each rating. The graph displays a multimodal distribution with peaks around ratings 3, 4, and 5. The mean, median, and mode values are provided, indicating a slightly right-skewed distribution with the mode at 4.0.

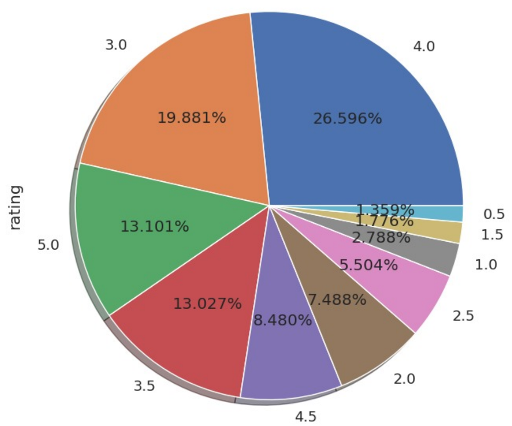

Figure 2: Pie Chart of Rating Variable

This pie chart visualizes the distribution of data across different rating values. The largest slices represent ratings 4.0 and 3.0, followed by 3.5, 5.0 and smaller proportions for other ratings like 2.5, 2.0, and 1.5. This chart provides a clear breakdown of the percentage contribution of each rating value.

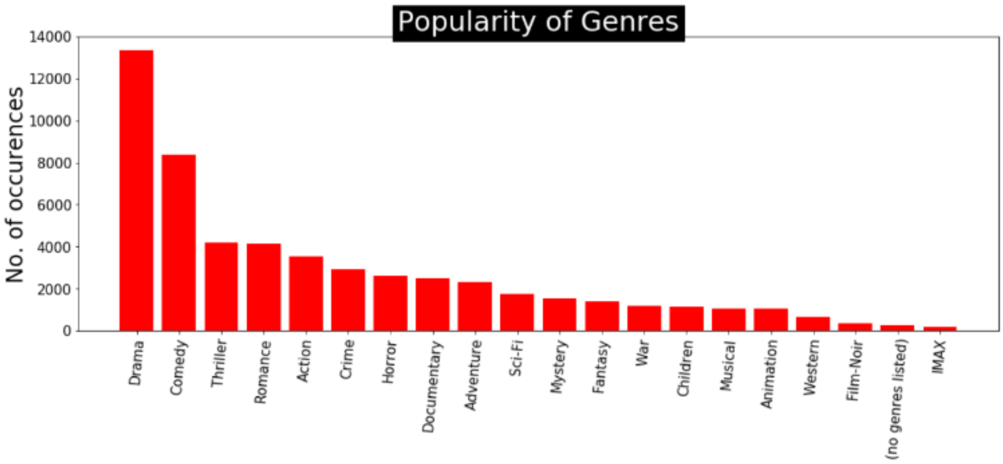

Figure 3: Popularity of various genres

This bar chart displays the popularity or frequency of different genres. The x-axis lists the genre names, and the y-axis represents the number of occurrences. The "Drama" genre has the highest bar, indicating its dominance, while genres like "Comedy," "Thriller," and "Romance" also have relatively high popularity. The remaining genres have considerably lower frequencies.

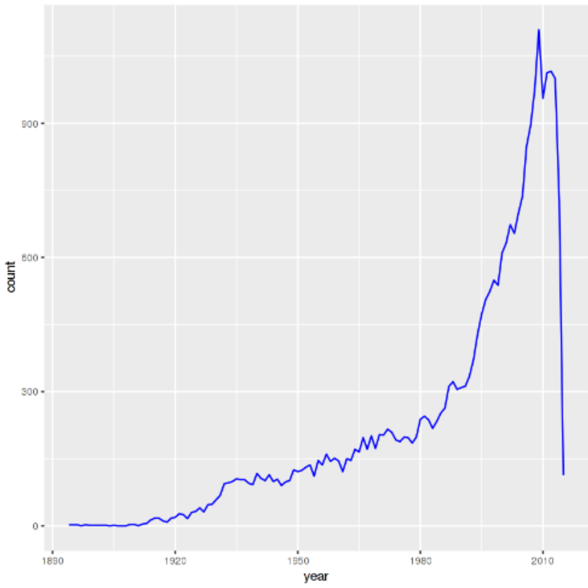

Figure 4: Year-wise distribution of movies

This line chart shows the trend in the number of movies released per year. The x-axis represents the year, and the y-axis displays the number of movies released in that year. The line starts relatively flat but exhibits a sharp upward trend in recent years, suggesting a rapid growth pattern in the movies released.

These visualizations provided valuable insights into the characteristics of the dataset, helping inform further analysis and modeling decisions in the movie recommendation system.

3.2 Results

We compared the various approaches that we tried by focusing on the root mean squared error and the mean squared error that they generated. The following tables display our findings.

Table 2: Results for traditional ML models

Model	**Mean Squared Error**
Random Forest Regressor	0.99
Gradient Boosting Regressor	1.01
Support Vector Regressor	1.02
Linear Regression	1.03
Ridge Regression	1.03

Table 3: Results for models implemented using Surprise

	KNN Basic	**SVD**	Baseline Only	Co-Clustering
RMSE	0.96	**0.88**	0.88	0.95
MSE	0.73	**0.67**	0.68	0.73

Clearly we can see that the SVD model implemented using the Surprise library outperforms all other approaches.

3.3 Example Inference

Below are a few inference examples which were generated using our movie recommendation system.

3.3.1 Content-Based Filtering

This example gives us a list of movies which are similar to the movie id 1, i.e, 'Toy Story (1995)'

```
Movies Similar to Movie with ID: 1

Antz (1998)
Toy Story 2 (1999)
Adventures of Rocky and Bullwinkle, The (2000)
Emperor's New Groove, The (2000)
Monsters, Inc. (2001)
Wild, The (2006)
Shrek the Third (2007)
Tale of Despereaux, The (2008)
Asterix and the Vikings (Asterix et les Vikings) (2006)
Turbo (2013)
```

We can see that the movies that are recommended are extremely similar to Toy Story. Toy Story which is an animated movie with the target demographic being young kids, and the recommended movies follow the same trend. The first recommended movie being 'Antz' which is an animated movie about ants. The second recommended movie is the sequel to Toy Story itself. We can conclude that the system gives satisfactory results.

3.3.2 Collaborative Filtering

This example gives us a list of movies personalised for the person with user id 1.

```
Top 10 Movie Recommendations for User 1:

Shawshank Redemption, The (1994)
Dr. Strangelove... (1964)
Lawrence of Arabia (1962)
Streetcar Named Desire, A (1951)
Departed, The (2006)
Dark Knight, The (2008)
Secrets & Lies (1996)
Rear Window (1954)
Guess Who's Coming to Dinner (1967)
Eternal Sunshine of the Spotless Mind (2004)
```

We can see from the dataset that the user has liked thriller and adventure movies and has watched and rated them higher than other genres, and the recommended movies also fall in the same genres.

4 Summary

This report introduced a Movie Recommendation System employing collaborative and content-based filtering techniques. Alongside our custom implementations, we utilized libraries like Surprise for collaborative filtering. Our system aims to deliver personalized movie suggestions to enhance user satisfaction.

Through experimentation, we demonstrated the effectiveness of our approach in providing accurate recommendations. By combining collaborative filtering, which analyzes user preferences based on historical ratings and similarities with other users, with content-based filtering, which recommends movies based on their attributes and user preferences, our system offers diverse and relevant recommendations.

Our findings indicate that personalized recommendations significantly impact user engagement and satisfaction. By tailoring suggestions to individual preferences, our system enhances the overall movie-watching experience. These results suggest potential benefits for movie platforms, including increased user retention and improved revenue generation.

References

[1] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. Research-paper recommender systems : a literature survey. *International Journal on Digital Libraries*, 17(4):305–338, 2016. ISSN 1432-5012. `doi:10.1007/s00799-015-0156-0`.

[2] Hrisav Bhowmick, Ananda Chatterjee, and Jaydip Sen. Comprehensive movie recommendation system, 2021.

[3] Nicolas Hug. Surprise: A python library for recommender systems. *Journal of Open Source Software*, 5(52):2174, 2020. `doi:10.21105/joss.02174`. URL `https://doi.org/10.21105/joss.02174`.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[5] Xun Zhou, Jing He, Guangyan Huang, and Yanchun Zhang. Svd-based incremental approaches for recommender systems. *Journal of Computer and System Sciences*, 81(4):717–733, 2015. ISSN 0022-0000. `doi:https://doi.org/10.1016/j.jcss.2014.11.016`. URL `https://www.sciencedirect.com/science/article/pii/S0022000014001706`.

A Contribution of each member

1. Maulik Desai: Contributed in model development, Visualization ,Final Report ,Mid progress Report, Spotlight Video
2. Abhinav Gupta: Contributed in model development ,Prepared Project Page
3. Tejas Gupta: Contributed in Model development,Visualization ,Mid progress Report,Deployed App on Stream lit, Spotlight Video
4. Harshit Yadav: Contributed in Data Cleaning and Visualization,Model development
5. Samay Mehar: Helped in Research and Mid progress Report
6. Mohit Kumar: Helped in Research