WATCHWISE-AI : Movie Rating Predictor & Recommender

A Project Report by :

Ananya Sharma – E23CSEU2227

Madhav Bhudhiraja – E23CSEU2238

SUBMITTED TO :

SCHOOL OF COMPUTER SCIENCE ENGINEERING AND TECHNOLOGY, BENNETT UNIVERSITY

GREATER NOIDA, 201310, UTTAR PRADESH, INDIA

April 2025

ABSTRACT

The Movie Recommendation System is designed to suggest movies to users based on different recommendation strategies. The system utilizes three distinct approaches: Popularity-Based Filtering, Content-Based Filtering, and Collaborative Filtering.

Popularity-Based Filtering recommends movies that have high average ratings and sufficient number of user ratings. Content-Based Filtering analyzes movie features to recommend similar movies based on genres and content attributes. Collaborative Filtering, implemented using the Surprise library, predicts user preferences by learning patterns from historical user-item interactions using models like SVD, User-Based CF, and Item-Based CF.

The dataset used is the MovieLens Latest Small Dataset, which provides user ratings and movie metadata. The models were trained and evaluated in Python, and the final application was developed using Streamlit, providing an interactive web interface for users to predict ratings and receive movie recommendations.

This project demonstrates the effectiveness of machine learning techniques in building personalized recommendation systems and offers a foundation for further enhancement into hybrid models.

INTRODUCTION

In today's digital era, recommendation systems play a crucial role in enhancing user experience by suggesting relevant items based on their interests. One of the most popular applications of recommendation systems is in the entertainment industry, especially for movies. With an ever-growing database of films, users often find it challenging to decide what to watch next. This project, titled **Movie Recommendation System**, aims to address this challenge by providing personalized movie suggestions to users.

The system leverages three main types of recommendation techniques:

- **Popularity-Based Filtering:** Recommends movies that are globally popular based on their average ratings and number of votes.

- **Content-Based Filtering:** Suggests movies similar to a selected movie by analyzing movie features like genre, keywords, and other attributes.
- **Collaborative Filtering:** Predicts a user's interest based on the behavior and preferences of other users. This approach uses models like SVD (Singular Value Decomposition), User-Based Collaborative Filtering, and Item-Based Collaborative Filtering, built using the Surprise library.

The project uses the **MovieLens Latest Small Dataset**, a well-known dataset in the research community for building and evaluating recommendation systems. Python is used for model development, with libraries like scikit-learn, scikit-surprise, pandas, and numpy. The final system is deployed through a web application built using **Streamlit**, providing an easy-to-use and interactive interface for end users.

This project not only explores fundamental recommendation algorithms but also showcases the practical deployment of machine learning models in real-world applications.

RELATED WORKS

Movie recommendation systems have evolved significantly over the years, driven by contributions from both researchers and industry leaders.

- **Netflix and the Netflix Prize:**
  Netflix has been a pioneer in advancing recommendation system technologies. In 2006, Netflix launched the *Netflix Prize* competition, offering a $1 million reward to the team that could improve their movie recommendation accuracy by 10%. This competition led to the development of innovative algorithms, especially around **Matrix Factorization** techniques like **SVD (Singular Value Decomposition)**, drastically improving collaborative filtering models.

- **GroupLens Research:**
  GroupLens, a research lab at the University of Minnesota, introduced the widely used *MovieLens datasets* and conducted extensive research on collaborative filtering methods. Their work has been foundational in the

recommender systems field, providing open datasets and benchmarking studies.

- **Pazzani and Billsus (1997):**
  They introduced early models for **content-based recommendation**, focusing on how systems could recommend new items (like movies) based on analyzing the features of items a user previously liked.

- **Burke (2002):**
  Burke proposed **hybrid recommender systems** that combine collaborative and content-based methods. His work addressed key challenges like the cold-start problem and improved recommendation robustness.

- **Amazon's Recommendation Engine:**
  Amazon popularized **item-based collaborative filtering** in production systems. Their recommendation algorithms focus on finding items similar to those previously bought or liked by a user, offering both scalability and personalization at a large scale.

PROBLEM STATEMENT

In today's digital age, users are presented with an overwhelming number of movie options across various streaming platforms. Without intelligent filtering and personalized suggestions, users often experience difficulty in selecting movies they are likely to enjoy.
Traditional browsing methods, such as genre or popularity lists, are insufficient to capture individual user preferences and lead to a suboptimal user experience.

There is a need for an efficient, scalable, and accurate movie recommendation system that can:

- Predict the ratings a user would likely assign to unseen movies.

- Recommend movies personalized to the user's tastes based on their past behavior and movie features.
- Address challenges like the cold-start problem, data sparsity, and scalability as the number of users and movies grow.

The aim of this project is to develop a movie recommendation system that integrates multiple recommendation strategies, including **popularity-based**, **content-based**, and **collaborative filtering models** (SVD, user-based CF, item-based CF), to deliver more accurate and diverse suggestions for users.

OBJECTIVES

The primary objective of this project is to design and implement a hybrid movie recommendation system that enhances the user experience by providing personalized and accurate movie suggestions.
The key goals include:

- **Develop a Popularity-Based Recommender:**
  Suggest top-rated movies universally liked by the majority of users.
- **Implement a Content-Based Filtering System:**
  Recommend movies similar to a selected movie based on content features like genres and tags, using cosine similarity.
- **Build Collaborative Filtering Models:**
  - **SVD (Singular Value Decomposition)** to predict user-movie ratings by learning latent factors.
  - **User-Based Collaborative Filtering** to recommend movies based on similar user profiles.
  - **Item-Based Collaborative Filtering** to recommend movies based on item similarity.
- **Preprocess and Prepare the Dataset:**
  Clean, transform, and feature-engineer the MovieLens dataset for optimal model performance.
- **Deploy the Recommendation System:**
  Create an interactive web application using Streamlit where users can select models, input their details, and receive personalized recommendations.
- **Ensure Model Interpretability and Usability:**
  Make the system easy to use and understand for both technical and non-technical users.

SCOPE OF PROJECT

This project focuses on building a **hybrid movie recommendation system** combining different recommendation strategies to offer a more personalized user experience.
The scope includes:

- **Dataset Used:**
  The MovieLens Latest Small dataset, containing user ratings, movie metadata like titles, genres, and IDs.
- **Recommendation Techniques Implemented:**
  - **Popularity-Based Filtering**: Recommending universally popular movies based on average ratings and minimum rating counts.
  - **Content-Based Filtering**: Using cosine similarity on movie features (genres, tags) to recommend similar movies.
  - **Collaborative Filtering**:
    - **SVD Model** for matrix factorization-based recommendations.
    - **User-Based Collaborative Filtering** to recommend movies liked by users with similar tastes.
    - **Item-Based Collaborative Filtering** to suggest movies similar to those a user has liked.
- **Technology Stack:**
  Python, Pandas, Scikit-learn, Scikit-Surprise, Streamlit for app development and deployment.
- **Deployment:**
  The system is designed to be deployed on cloud platforms like Render to make it accessible via a web interface.
- **Target Users:**
  Movie enthusiasts seeking new movie recommendations based on their preferences and history.
- **Limitations:**
  - Model performance can vary based on the size and quality of the dataset.
  - Cold-start problem (new users or new movies) is not fully addressed.
  - Real-time user behavior updates are not incorporated.

TECHNOLOGIES USED

Technologies and Models

Technologies Used

Category	Tools / Libraries
Programming Language	Python
Data Manipulation	Pandas, NumPy
Machine Learning Models	Surprise (SVD, KNNBasic), Scikit-learn
Similarity Calculation	cosine_similarity* from sklearn metrics pairwelse
Model Saving/Loading	joblib
Web Framework	Streamlit
Data Visualization (Optional)	Matplotlib, Seaborn (Optional)
IDE/Notebook	Google Colab / VS Code
Dataset	MovieLens Latest Small Dataset

Models Used

Modal Type	Model Name	Description
Collaborative Filtering	SVD	Matrix factorlization technique from Surprise to predict ratings
User-Based CF	User-Based CF KNN-based similarity between users	KNN-based similarity between users
Content-Based Filtering	Cosine Similarity	Measures similarity between movie feature vectors
Popularity-Based Filtering	Mean Rating + Count)	Non-persoml-ized, recommerds top-rated movies

CONTRIBUTIONS

This project presents a comprehensive movie recommendation system that integrates multiple recommendation approaches to enhance user experience. The key contributions of this project are:

1. Multi-Model Recommendation System

- Designed and implemented **five types of recommenders**:
  - Popularity-Based Recommendation
  - Content-Based Recommendation

- Collaborative Filtering using SVD
- User-Based Collaborative Filtering
- Item-Based Collaborative Filtering

- Gave users flexibility to choose between different recommendation techniques based on their preferences.

2. Integration of Machine Learning Models

- Successfully trained and saved machine learning models using **Surprise** and **Scikit-learn** libraries.
- Utilized matrix factorization (SVD) and similarity-based algorithms to accurately predict user ratings.

3. End-to-End Deployment

- Built an **interactive web application** using **Streamlit** that allows users to predict ratings and get personalized movie recommendations.
- Prepared the project for **deployment on Render** with properly structured requirements.txt and runtime.txt files for smooth environment setup.

4. User-Friendly Interface

- Created a clean, simple, and interactive interface where:
  - Users can input their user ID.
  - Select a movie from a dropdown.
  - Instantly get predicted ratings and top recommended movies.

5. Handling Data Challenges

- Carried out **data cleaning, feature engineering, and similarity computation** to create meaningful movie feature vectors.
- Managed missing data, inconsistencies, and ensured robust similarity calculations for content-based filtering.

6. Research and Learning

- Studied and implemented state-of-the-art recommendation techniques.
- Understood practical challenges of model training, saving/loading, and web deployment in a real-world setting.

CONCLUSION

In this project, I developed a movie recommendation system using different techniques, including content-based filtering, collaborative filtering, and popularity-based methods. The system leverages the MovieLens dataset and utilizes models such as cosine similarity, collaborative filtering via the Surprise library (SVD, user-based, and item-based), and a simple popularity-based approach.

Throughout the project, I performed extensive data preprocessing, including handling missing values, feature engineering, and exploratory data analysis to gain valuable insights into the data. I then applied the various recommendation algorithms, ensuring to evaluate and compare their performance using metrics such as accuracy and user satisfaction.

The results demonstrated that the content-based filtering model, utilizing cosine similarity, performed well for suggesting similar movies based on the features, while collaborative filtering methods (SVD and user/item-based) were more effective for personalized recommendations based on user behavior.

This project has been a valuable learning experience in implementing machine learning models and deploying them on a web-based platform using Streamlit. The movie recommendation system can be further enhanced by incorporating additional features such as hybrid models combining both content-based and collaborative methods, or by including more sophisticated approaches like deep learning models.

Finally, this system can serve as a foundation for more complex recommendation engines and be expanded to support real-time recommendations for users based on their interaction with the platform.

FUTURE SCOPE

While the current movie recommendation system provides valuable suggestions to users based on multiple algorithms, there are several areas where the system can be improved and extended:

1. **Hybrid Recommendation System**: Combining content-based filtering and collaborative filtering models could yield more accurate and personalized recommendations. This hybrid approach can balance the strengths and weaknesses of each individual model.
2. **Incorporating Deep Learning**: Future versions of the system can explore deep learning models, such as neural collaborative filtering (NCF) or autoencoders, to capture more complex patterns and provide even more accurate recommendations.
3. **Real-Time Recommendations**: The system could be expanded to provide real-time recommendations, adjusting dynamically based on a user's most recent interactions or preferences.
4. **Handling Cold Start Problem**: Incorporating better solutions for the cold start problem (i.e., recommending movies to new users or recommending new movies with limited data) would improve the system's effectiveness for all users.
5. **User Feedback Integration**: Allowing users to rate the recommendations and providing a feedback loop could help refine the recommendations further, creating a more personalized user experience.
6. **Scalability**: As the dataset grows, the system can be optimized for scalability, ensuring that it performs well even with large datasets and increasing user activity.
7. **Integration with Other Data Sources**: The system can be enhanced by incorporating additional datasets, such as IMDb ratings or movie reviews, to improve the depth of the recommendations.

By implementing these improvements, the recommendation system could evolve into a more sophisticated and robust platform, capable of providing highly personalized and efficient movie recommendations.

REFERENCES

1. **MovieLens Dataset**. (n.d.). *GroupLens Research*. Retrieved from https://grouplens.org/datasets/movielens/
2. **Surprise Library**. (n.d.). *Python package for building and analyzing recommender systems*. Retrieved from https://surprise.readthedocs.io/en/stable/
3. **Streamlit**. (n.d.). *Streamlit: The fastest way to build and share data apps*. Retrieved from https://streamlit.io/
4. **Scikit-learn**. (n.d.). *Machine Learning in Python*. Retrieved from https://scikit-learn.org/
5. **KNN Algorithm**. (n.d.). *K-Nearest Neighbors Algorithm for Classification and Regression*. Retrieved from https://towardsdatascience.com/k-nearest-neighbors-knn-for-machine-learning-346db6a3ad1d
6. **Cosine Similarity**. (n.d.). *A simple guide to cosine similarity*. Retrieved from https://towardsdatascience.com/a-simple-guide-to-cosine-similarity-2d1281a4f94f
7. **Movie Recommendation Systems: A Survey and Future Directions**. (2018). *International Journal of Advanced Computer Science and Applications*. Retrieved from https://thesai.org/Downloads/Volume9No6/Paper_22-Movie_Recommendation_Systems.pdf
8. **Collaborative Filtering and Content-Based Filtering**. (n.d.). *Recommender Systems and Their Algorithms*. Retrieved from https://www.analyticsvidhya.com/blog/2020/08/a-complete-guide-to-understand-collaborative-filtering-and-content-based-filtering/

THANK YOU