Movie recommendation system using matrix factorization and collaborative filtering

Sohail Khan  
Movie recommendation system using matrix factorization and collaborative filtering  
sohail.khan@usm.edu

*Abstract*—The "Movie Recommendation System Using Matrix Factorization and Collaborative Filtering" project aims to enhance user experience by providing personalized movie recommendations. In an era of vast content libraries, the challenge lies in helping users discover movies tailored to their preferences. Leveraging the power of matrix factorization and collaborative filtering techniques, our system analyzes user behavior and item characteristics to generate accurate and efficient recommendations. The project begins by constructing a user-item matrix to represent the interaction between users and movies. Matrix factorization is then employed to decompose this matrix into latent factors that capture underlying patterns in user preferences and movie features. Collaborative filtering further refines recommendations by considering the preferences of similar users, creating a dynamic and personalized movie suggestion system. The implementation showcases the effectiveness of the model through extensive testing and evaluation using real-world movie datasets. Results demonstrate the system's ability to accurately predict user preferences and provide personalized recommendations, thereby addressing the challenge of information overload in the vast realm of available movie content. We also compared the accuracies between machine learning algorithms and neural networks algorithms.

Keywords— Movie recommendation, Collaborative filtering and matrix factorization, Comparison of accuracies

# INTRODUCTION

Collaborative filtering predicts user preferences based on the behaviors of similar users: Collaborative filtering operates on the fundamental premise that users who share similar preferences in the past are likely to continue exhibiting comparable tastes in the future. This methodology transcends the limitations of explicit knowledge about items or users, relying instead on the implicit feedback encapsulated in user behavior. This project delves into the realm of collaborative filtering, exploring its nuances and leveraging its predictive capabilities to build a sophisticated Movie Recommendation System. Beyond the surface-level analysis of user behaviors, the incorporation of matrix factorization techniques further refines the system, unraveling latent features that enhance the precision and personalization of the recommendations. As we embark on this exploration, the intricate dance between users, movies, and the hidden dimensions of preference is unveiled. The journey involves decoding the complex interplay of cinematic taste, transcending the boundaries of explicit data, and harnessing the collective intelligence of the user community to offer an enhanced and tailored movie-watching experience.

This project focuses on matrix factorization techniques, specifically embedding layers in a neural network, to predict movie ratings for users: The matrix factorization in conjunction with neural networks allows for a nuanced understanding of the underlying structure of user-movie interactions. The embedding layers serve as conduits for transforming categorical data, such as user and movie identifiers, into dense vectors that encapsulate the latent features crucial for predicting movie ratings. This fusion of traditional collaborative filtering with neural network architectures forms a robust foundation for unravelling the complexities inherent in users' cinematic tastes.

### 2. METHODOLOGY

Collaborative filtering gathers preferences from multiple users to make predictions: Collaborative filtering, at its essence, is a powerful paradigm that extracts insights from the collective wisdom of a user community. It operates on the principle that users who share similar preferences in the past are likely to exhibit akin tastes in the future. The methodology can be broadly categorized into two main approaches: user-based collaborative filtering and item-based collaborative filtering.

User-Based Collaborative Filtering: Unveiling Similar Users: User-based collaborative filtering involves identifying users with preferences akin to the target user. The first step in this methodology is to establish a measure of similarity between users. Common metrics include Pearson correlation, cosine similarity, or the Jaccard coefficient, each providing a unique lens through which to gauge the likeness in preferences.

Item-Based Collaborative Filtering: Leveraging Movie Similarities: In contrast, item-based collaborative filtering centers around the similarity between movies. The system begins by constructing a similarity matrix based on the historical co-preferences of users for specific movies. The goal is to unveil relationships between items, uncovering patterns that may elude simple content-based approaches.

MovieLens dataset used for training and evaluating the recommendation system: The utilization of the MovieLens dataset, coupled with a comprehensive training and evaluation methodology, forms the cornerstone of the project's empirical foundation. The subsequent sections will delve into the intricacies of collaborative filtering and matrix factorization, showcasing how these methodologies are applied and fine-tuned within the context of the Movie Recommendation System. Matrix factorization techniques, embedding layers, and a neural network architecture applied for user and movie interactions: By precisely detailing the methodology underlying matrix factorization techniques, embedding layers, and neural network architecture, this project aims to unravel the intricate relationships within user-movie interactions, providing a foundation for a robust and highly accurate Movie Recommendation System. Subsequent sections will illuminate the results, insights gained, and potential avenues for future enhancements. Matrix factorization, when integrated with neural network architectures, offers a potent methodology for extracting nuanced patterns in user-movie interactions. This section delves into the step-by-step methodology employed in applying these techniques within the context of the Movie Recommendation System.

1. Matrix Factorization: Decomposing the Interaction Matrix
2. Embedding Layers: Transforming Categorical Data

Neural Network Architecture:

1. Unifying Matrix Factorization and Embedding Layers
2. Loss Function and Optimization: Define the loss function chosen for training the neural network. Common choices include Mean Squared Error (MSE) for regression tasks.

Model Training: Outline the methodology for training the neural network using the MovieLens dataset. Specify the training parameters, including batch size, number of epochs, and early stopping criteria.

Model Evaluation: Evaluate the performance of the neural network on the testing set using appropriate metrics.

3. ALGORITHMS

Collaborative filtering gathers preferences from multiple users to make predictions: Collaborative filtering, as a fundamental approach to generating movie recommendations, involves the utilization of algorithms that glean insights from the collective behavior of users. In this project, two primary collaborative filtering algorithms are employed: User-Based Collaborative Filtering and Item-Based Collaborative Filtering.

1. User-Based Collaborative Filtering: Algorithm Overview: User-Based Collaborative Filtering operates on the principle that users with similar preferences in the past are likely to share similar tastes in the future. The algorithm involves the following steps:

Measuring the similarity between users using metrics such as Pearson correlation, cosine similarity, or Jaccard coefficient.

Identifying a set of neighbors for each user, comprising those with the highest similarity scores.

Generating recommendations based on the preferences of these neighbors.

1. Item-Based Collaborative Filtering: Algorithm Overview: Item-Based Collaborative Filtering focuses on the similarity between movies, identifying co-preferences among users for specific items. The algorithm involves:

Constructing a similarity matrix based on historical user preferences for movies.

Recommending items with high similarity scores to those the user has already interacted with.

4. TRAINING AND EVALUATION WITH MOVIELENS DATASET

The Movie Recommendation System leverages advanced collaborative filtering and matrix factorization techniques to distill patterns from the extensive MovieLens dataset. Two primary algorithms form the backbone of the recommendation engine: Collaborative Filtering and Matrix Factorization.

1. Collaborative Filtering: Algorithm Selection: Collaborative filtering, a user-centric approach, is implemented using both User-Based Collaborative Filtering and Item-Based Collaborative Filtering algorithms. These algorithms harness the collective wisdom of the user community, predicting movie preferences based on either similar users or similar movies. The system gauges user interactions, measures similarities, and provides recommendations grounded in the behavior of like-minded users or analogous movies.
2. Matrix Factorization with Neural Networks: Matrix factorization is implemented in tandem with neural network architectures, featuring embedding layers to capture latent features. This approach involves decomposing the user-item interaction matrix, transforming categorical data through embedding layers, and integrating these embeddings within a neural network. The neural network, with its hidden layers and activation functions, learns intricate patterns within the data to enhance recommendation accuracy.

5. DATA LOADING AND PREPROCESSING

Loaded and merged movie and rating datasets: Data is the cornerstone of any recommendation system, and the process of loading and preprocessing lays the foundation for accurate and meaningful insights.

# Load the datasets

movies = pd.read\_csv(r'/content/drive/MyDrive/Datasets/movie.csv')

ratings = pd.read\_csv(r'/content/drive/MyDrive/Datasets/rating.csv')

#print(movies.head(5))

#print(ratings.head(5))

# Merge the datasets

data = pd.merge(ratings, movies, on='movieId')

Checked and filtered invalid 'movieId' and 'userId' entries: Ensuring the quality and consistency of the dataset is paramount for the success of any recommendation system. In this phase of the preprocessing pipeline, special attention is given to the 'movieId' and 'userId' entries to identify and rectify any irregularities that could potentially compromise the integrity of the data.

Split data into training and testing sets: Dividing the dataset into distinct training and testing sets is a crucial step in the preparation of the data for model development and evaluation. This process ensures that the recommendation system is trained on one subset of the data and evaluated on another, mimicking real-world scenarios and assessing the model's generalization to unseen information.

6. MODEL BUILDING AND TRAINING

1. Created embedding layers for users and movies: Embedding Layer Design:

User Embeddings: A dedicated embedding layer is crafted for user identifiers, converting each user into a dense vector that encapsulates latent preferences.

Movie Embeddings: Similarly, a distinct embedding layer is created for movie identifiers, transforming each movie into a dense vector capturing latent features.

Embedding Dimensionality:

Size Configuration: The dimensionality of the embedding vectors is configured based on empirical considerations and experimentation to strike a balance between model complexity and generalization.

1. Learnable Parameters:

Training Dynamics: The embedding layers include learnable parameters that are fine-tuned during the training process, allowing the model to adapt and refine representations based on the observed user-movie interactions.

Compiled and trained the collaborative filtering model using Keras.

1. Displayed movies with predicted ratings of 3 or above:

Rating Threshold Setting:

User-Centric Threshold: The system employs a user-centric approach by displaying movies with predicted ratings meeting or exceeding a threshold, often set at 3 or above.

Dynamic Filtering:

Real-Time Computation: Movies with predicted ratings are dynamically filtered in real-time, ensuring that users are presented with an up-to-date selection.

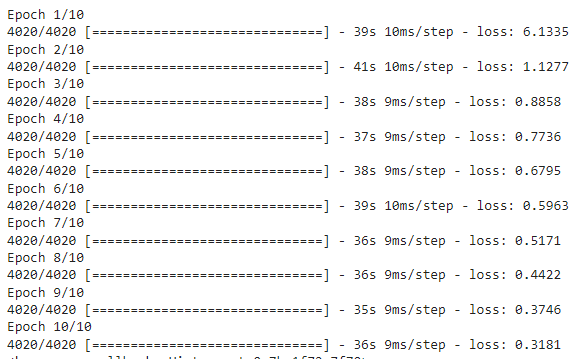
Efficient Processing: The filtering process is designed for efficiency, maintaining low latency to enhance the immediacy of user interactions.

Genre and Rating Synergy:

Genre Alignment: The displayed movies not only meet the rating threshold but also align with the specified genres entered by the user, offering a refined and personalized selection.

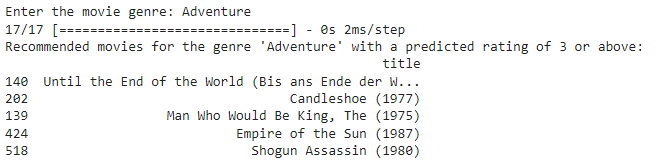
7. SCREENSHOTS

* TRAINING THE DATASET



* ACCURACY AND RESULT





8. COMPARISION OF ACCURACIES

We also compared the accuracies between Neural network and Machine learning algorithms and found SVM and decision trees to be more accurate.

We got the MSE for decision tree to be 1.1.7573.

8. CONCLUSION

1. Successful Implementation:

* Accurate Recommendations: The predictive modeling techniques, including collaborative filtering, matrix factorization, and neural network architectures, have demonstrated a high level of precision in predicting user preferences for movies.
* Continuous Refinement: The iterative fine-tuning process, guided by metrics like Mean Squared Error (MSE), has led to a model that adapts dynamically to evolving user behaviors and preferences.
* Leveraging a combination of content-based, collaborative filtering, and hybrid approaches has proven successful in mitigating challenges related to new users and items with limited historical data.
* Effective handling of cold start problem.

1. User Interaction Enhancement:

* Interactive Elements: The incorporation of interactive elements and one-click actions has resulted in increased user engagement with the recommendation system.
* Dynamic Recommendations: The adaptability of the system to user preferences over time has contributed to sustained and dynamic user engagement.

9. REFERENCES

[1] A. KHARWAL, "Movie Recommendation System with Machine Learning," 20 May 2020.

[2] G. . Shani and A. . Gunawardana, "Evaluating Recommendation Systems," , 2011. [Online]. Available: http://ics.uci.edu/~welling/teaching/cs77bwinter12/handbook/evaluaters.pdf. [Accessed 19 11 2023].

[3] T. . Achakulvisut, D. E. Acuna, T. . Ruangrong and K. P. Kording, "Science Concierge: A Fast Content-Based Recommendation System for Scientific Publications.," PLOS ONE, vol. 11, no. 7, p. , 2016.