AI-Based Movie Recommendation System

Abstract

This paper presents a movie recommendation system inspired by popular platforms like Netflix and Amazon, leveraging advanced machine learning techniques. The system utilizes collaborative filtering based on user ratings to predict movies a user might like. It employs cosine similarity and matrix factorization for generating personalized recommendations. The project was implemented in Python using tools such as pandas, numpy, scikit-learn, and a dataset sourced from IMDb. The goal of this research is to provide an intuitive and scalable recommendation system that delivers a personalized user experience.

---

1. Introduction

With the explosion of online streaming services, recommendation systems have become an essential tool for providing personalized content to users. Movie recommendation systems are designed to help users find movies they are likely to enjoy based on past behavior and preferences. Netflix, Amazon, and other platforms have widely adopted these techniques to enhance user satisfaction and retention.

In this project, we focus on a collaborative filtering approach, which leverages user-item interactions (in this case, movie ratings) to recommend content. Collaborative filtering is advantageous because it doesn't require explicit metadata about movies but instead relies on patterns between users and items. Our system was built using Python and relies on machine learning libraries such as pandas, numpy, and scikit-learn.

---

2. Problem Definition

The challenge is to predict a list of movies that a user is likely to enjoy based on their historical ratings and those of similar users. Given a dataset of user ratings, we aim to predict a user's rating for unrated movies and recommend the top-rated ones.

---

3. Mathematical Foundation

The core of the recommendation engine is **collaborative filtering**. Collaborative filtering predicts what a user might like based on the preferences of other similar users. There are two main approaches:

3.1. User-Based Collaborative Filtering:

We calculate the similarity between users and use this similarity to predict the ratings for items that a user has not rated yet. The similarity between two users uuu and vvv is calculated using **cosine similarity**, defined as:

$$\text{cosine\_similarity}(u, v) = \frac{\sum_{i \in I} r_{u,i} \cdot r_{v,i}}{\sqrt{\sum_{i \in I} r_{u,i}^2} \cdot \sqrt{\sum_{i \in I} r_{v,i}^2}}$$

Where:

- ru,ir_{u,i}ru,i is the rating user uuu gave to movie iii,
- III is the set of movies rated by both users uuu and vvv.

3.2. Item-Based Collaborative Filtering:

Alternatively, we can use item similarity to predict a user's rating for unrated items. The similarity between items iii and jjj is calculated as:

$$\text{cosine\_similarity}(i, j) = \frac{\sum_{u \in U} r_{u,i} \cdot r_{u,j}}{\sqrt{\sum_{u \in U} r_{u,i}^2} \cdot \sqrt{\sum_{u \in U} r_{u,j}^2}}$$

Where:

- ru,ir_{u,i}ru,i is the rating given by user uuu to movie iii,
- UUU is the set of users who rated both items iii and jjj.

3.3. Matrix Factorization:

We also explore **Matrix Factorization** techniques such as Singular Value Decomposition (SVD) to factorize the user-item rating matrix into lower-dimensional matrices. These matrices represent latent features that capture the underlying structure of user preferences and item characteristics.

Given a user-item matrix RRR, SVD decomposes it as:

$$R \approx U \cdot S \cdot V^T$$

Where:

- UUU is the matrix of user features,
- SSS is the diagonal matrix of singular values (latent factors),
- VTV^TVT is the transpose of the movie feature matrix.

---

4. Methodology

4.1. Data Collection

We used the IMDb dataset for this project. The dataset includes user ratings for various movies. The main data includes:

- **Movies dataset**: Contains information about movies (ID, title).
- **Ratings dataset**: Contains user IDs, movie IDs, and ratings given by users.

4.2. Preprocessing

We cleaned and formatted the data for ease of use:

- Removing duplicates and missing values.
- Normalizing ratings.
- Splitting the dataset into training and testing sets.

4.3. Building the Model

The core of our system is based on collaborative filtering. We implemented both user-based and item-based collaborative filtering approaches using cosine similarity to calculate the similarity between users and movies.

For matrix factorization, we used SVD from the scikit-learn library to reduce the dimensionality of the user-item matrix and improve recommendation accuracy.

4.4. Model Evaluation

We evaluated our system by splitting the dataset into training and testing subsets, then calculating the Root Mean Square Error (RMSE) to measure the accuracy of the predicted ratings.

6. Results

The system was tested on the IMDb dataset, providing accurate recommendations for users based on their past behavior. The cosine similarity and matrix factorization approaches both showed strong performance, with matrix factorization being slightly more efficient in cases with sparse data.

---

7. Conclusion

This paper presents a movie recommendation system built using collaborative filtering and matrix factorization techniques. The system is highly scalable and can be applied to a variety of recommendation tasks. Future improvements could involve integrating additional metadata such as movie genres, descriptions, and user demographics to further enhance recommendation quality.

---

8. References

1. Ricci, F., Rokach, L., & Shapira, B. (2011). **Introduction to Recommender Systems Handbook**.
2. Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems.