Movie Recommendation System using Machine Learning

Abstract

This project focuses on building an intelligent Movie Recommendation System using Machine Learning techniques. The system suggests movies to users based on their preferences by combining collaborative filtering and content-based filtering. The application is implemented using Python, Pandas, Scikit-learn, and Streamlit, and utilizes the MovieLens dataset for training and evaluation.

1. Introduction

With the rapid growth of online streaming platforms, recommendation systems have become essential to enhance user experience. A Movie Recommendation System helps users discover relevant movies by analyzing past behavior, preferences, and item characteristics. This project aims to design and deploy a recommendation engine that provides personalized movie suggestions.

2. Objectives

- To build a movie recommendation system using Machine Learning techniques.
- To implement collaborative filtering based on user ratings.
- To integrate content-based filtering using movie genres.
- To design an interactive Streamlit web interface.
- To provide top 5 personalized movie recommendations.

3. Dataset Description

The MovieLens dataset is used for this project. It contains information about movies, user ratings, and genres. The primary files used include movies.csv (movieId, title, genres) and ratings.csv (userId, movieId, rating, timestamp). This dataset is widely used for benchmarking recommendation algorithms.

4. Methodology

4.1 Data Preprocessing

Data preprocessing includes handling missing values, merging datasets, encoding genres, and filtering active users and movies. The ratings data is transformed into a user-item interaction matrix for collaborative filtering.

4.2 Collaborative Filtering

Collaborative filtering is implemented using cosine similarity on the user-item matrix. This approach recommends movies by identifying users with similar rating patterns.

4.3 Content-Based Filtering

Content-based filtering uses movie genres to compute similarity between movies. TF-IDF vectorization and cosine similarity are applied to recommend movies similar to a user’s preferred genres.

4.4 Hybrid Recommendation Approach

A hybrid model combines collaborative and content-based filtering to improve recommendation accuracy and handle cold-start problems.

5. System Architecture

The system consists of a data layer (MovieLens dataset), a processing layer (ML models and similarity computation), and a presentation layer (Streamlit web application). User inputs are processed in real time to generate recommendations.

6. Tools and Technologies Used

- Python – Core programming language
- Pandas & NumPy – Data manipulation and analysis
- Scikit-learn – Machine learning algorithms and similarity metrics
- Streamlit – Web application development
- MovieLens Dataset – Training and evaluation data

7. Results and Output

The system successfully generates the top 5 personalized movie recommendations based on user preferences. The Streamlit interface allows users to select movies or genres and instantly view recommended results.

8. Future Enhancements

- Incorporate sentiment analysis using movie reviews.
- Use deep learning-based recommendation models.
- Add user authentication and profile management.
- Deploy the application on cloud platforms.

9. Conclusion

This project demonstrates the effective use of Machine Learning techniques to build a Movie Recommendation System. By combining collaborative and content-based filtering, the system provides accurate and personalized recommendations. The project is suitable for academic learning as well as real-world applications.