**Aim**

To develop a Movie Recommendation System that suggests personalized movie recommendations to users based on their preferences and interaction history using machine learning algorithms.

**📚 Theory**

A **Movie Recommendation System** is a subclass of information filtering systems that seek to predict the "rating" or "preference" a user would give to a movie. There are three major types of recommendation approaches:

1. **Content-Based Filtering**:
   * Recommends movies similar to those the user liked.
   * Based on movie features (genre, actors, language, etc.).
2. **Collaborative Filtering**:
   * Recommends movies liked by similar users.
   * Finds user-user or item-item similarities.
3. **Hybrid Filtering**:
   * Combines both content-based and collaborative filtering for better accuracy and personalization.

The system uses machine learning models to learn user behavior and patterns from the dataset and applies that knowledge to provide meaningful recommendations.

**🧩 Class Diagram**

Below is a simple representation of a class diagram for the Movie Recommendation System:

pgsql

CopyEdit

+----------------+ +------------------+ +----------------+

| User | | Movie | | Rating |

+----------------+ +------------------+ +----------------+

| - userId | | - movieId | | - ratingId |

| - name | | - title | | - userId |

| - email | | - genre | | - movieId |

| - preferences | | - releaseYear | | - rating |

+----------------+ | - language | | - timestamp |

+------------------+ +----------------+

| | |

| 1 | 1 | \*

|--------------------------|-------------------------|

**🔍 Explanation:**

* **User**: Represents each user of the system.
* **Movie**: Contains all movie details like title, genre, year, etc.
* **Rating**: A relationship class where each user rates a movie.

This design helps in easily applying collaborative filtering using the userId and movieId to generate recommendations.

**📝 Conclusion**

The Movie Recommendation System effectively utilizes user data and movie metadata to deliver personalized recommendations. By implementing algorithms like collaborative filtering and content-based filtering, the system enhances user engagement and satisfaction. Through proper modeling, such as class diagrams, we ensure a structured and scalable design for real-world deployment.