**Task 5: Movie Recommendation System**

**1. Objective**

The goal of this task is to design a **recommendation system** that suggests movies to users based on their interests and preferences.  
Recommendation systems play a crucial role in platforms like **Netflix, Amazon Prime, and IMDb**, where personalized suggestions enhance user engagement and satisfaction.

**2. Dataset**

* **Source**: Movie Recommendation Dataset (e.g., MovieLens dataset from Kaggle).
* **Dataset Size**: Typically includes thousands of movies and user ratings.
* **Main Features**:
  + userId – Unique identifier of the user.
  + movieId – Unique identifier of the movie.
  + title – Movie name.
  + genres – Movie categories (e.g., Action, Comedy, Drama).
  + rating – User-provided rating (scale 1–5).
  + timestamp – When the rating was given.

**3. Problem Type**

* **Unsupervised / Semi-supervised Learning**
* **Recommendation System**

**4. Approach**

**(a) Data Preprocessing**

1. **Data Cleaning** – Removed missing or invalid ratings.
2. **Encoding** – Handled categorical features like genres.
3. **Feature Engineering** – Created a **user-item rating matrix**.

**(b) Recommendation Techniques**

Two main approaches were implemented:

1. **Content-Based Filtering**
   * Uses **movie metadata** (genres, keywords, etc.) to recommend similar movies.
   * Example: If a user liked “Inception”, recommend other *sci-fi/thriller* movies.
2. **Collaborative Filtering (CF)**
   * Uses **user-item interactions**.
   * Based on the assumption: *“Users who liked similar movies in the past will like similar ones in the future.”*
   * Implemented using:
     + **User-User CF** – Recommend movies liked by similar users.
     + **Item-Item CF** – Recommend movies similar to those a user has already liked.

**(c) Advanced Approach (Bonus)**

* **Matrix Factorization (SVD)**: Decomposed user-item matrix into latent factors to capture hidden preferences.

**5. Evaluation Metrics**

* **RMSE (Root Mean Squared Error)** – To measure prediction accuracy of ratings.
* **Precision@K and Recall@K** – To evaluate recommendation relevance.
* **Top-N Recommendation Lists** – Tested with sample users.

**6. Visualization**

* Distribution of ratings (most users rate movies between 3–5).
* Popular movies (highest rated vs most rated).
* Sample recommendation lists for selected users.

**7. Tools & Libraries**

* **Python**
* **Pandas, NumPy** → Data preprocessing
* **Matplotlib, Seaborn** → Visualization
* **Scikit-learn** → Similarity measures (Cosine similarity, Nearest Neighbors)
* **Surprise library / SVD** → Collaborative filtering & matrix factorization

**8. Results**

* **Content-Based Filtering**: Provided relevant recommendations but limited to known genres.
* **Collaborative Filtering**: Captured user preferences better by analyzing rating patterns.
* **Matrix Factorization (SVD)**: Outperformed simple CF with higher accuracy and better personalization.

**9. Applications**

* **OTT Platforms**: Netflix, Prime Video, Disney+ for recommending movies/TV shows.
* **E-commerce**: Amazon-style product recommendation (similar logic).
* **Music & Books**: Spotify, Goodreads use similar algorithms.
* **Personalization**: Improves customer engagement and retention.



