# Movie Recommendation System

## 1. Project Overview

The Movie Recommendation System is designed to provide personalized movie recommendations based on user preferences and movie attributes. The project utilizes the TMDB 5000 Movies Dataset, leveraging various machine learning techniques to create an efficient and user-friendly recommendation engine. This system can recommend movies by genre, popularity, director, cast, and other features.

## 2. Objective

The primary goal of this project is to build a recommendation system that suggests movies similar to a user’s input or based on trends, using content-based filtering and collaborative filtering methods. The system aims to help users discover new content they are likely to enjoy, enhancing their overall movie-watching experience.

## 3. Dataset

• Name: TMDB 5000 Movies Dataset  
• Source: The Movie Database (TMDB)  
• Description: The dataset contains metadata for 5000 movies, including features such as:  
 - Movie title  
 - Genre  
 - Cast and Crew (e.g., actors, directors)  
 - Plot Keywords  
 - Release Date  
 - Popularity Score  
 - Vote Count and Average Rating  
  
This rich dataset allows for robust content filtering based on movie attributes, as well as collaborative filtering by user interactions.

## 4. Methodology

The project employs a hybrid recommendation approach, integrating Content-Based Filtering and Collaborative Filtering techniques.  
  
1. Content-Based Filtering:  
 - This method suggests movies similar to the one a user has previously liked or watched. It uses the metadata (genres, actors, directors, keywords) to calculate similarity scores.  
 - Cosine Similarity: Calculates the cosine of the angle between two vectors in a multi-dimensional space, determining how closely related two movies are based on their attributes.  
  
2. Collaborative Filtering:  
 - This approach is based on user interaction data, suggesting movies that similar users have enjoyed.  
 - Matrix Factorization Techniques: Using algorithms like SVD (Singular Value Decomposition) to predict user ratings and preferences.  
  
3. Implementation and System Workflow:  
 - Data Preprocessing: Cleaning and formatting the dataset for efficient use, including handling missing values, and transforming categorical data.  
 - Feature Extraction: Using natural language processing and vectorization techniques to convert textual data into numerical representations (e.g., TF-IDF).  
 - Model Training and Evaluation: Training machine learning models and evaluating their performance using metrics such as precision, recall, and F1 score.

## 5. Software and Libraries Used

• Python: Core programming language used for model development.  
• Jupyter Notebook: For exploratory data analysis and interactive code execution.  
• Pandas: Data manipulation and handling library, essential for preprocessing.  
• Numpy: Library for handling numerical operations and data structures.  
• Scikit-learn: Machine learning library used for similarity calculations and model evaluation.  
• NLTK and SpaCy: Natural Language Processing libraries used for text processing.  
• Flask: Web framework used to deploy the recommendation system as a web app.  
• SQLite/MongoDB: Database for storing movie data and user preferences.

## 6. Project Structure and Code Implementation

• Data Preprocessing Notebook (mrecom.ipynb):  
 - Prepares the dataset for modeling by cleaning and transforming data.  
 - Applies feature extraction techniques on genres, cast, crew, and keywords.  
 - Implements similarity calculations for Content-Based Filtering.  
  
• Application Code (apps.py):  
 - Contains the backend logic for the recommendation system.  
 - Implements functions for retrieving recommendations based on user input.  
 - Includes API endpoints to interact with the frontend, delivering recommendations in real-time.  
 - Manages database connections and fetches user and movie data.

## 7. User Interface

The web application is developed using Flask as a backend and simple HTML/CSS for the frontend. The UI allows users to search for a movie, view its details, and receive similar movie recommendations instantly.

## 8. Results and Evaluation

The recommendation system provides relevant and personalized movie suggestions based on both the attributes of movies and user interaction data.

## 9. Future Work

Future improvements to the system could include:  
• User-based Collaborative Filtering: Enhancing recommendations by incorporating user ratings and interactions.  
• Advanced NLP Techniques: Utilizing transformer models for better understanding movie descriptions and user reviews.  
• Integration with TMDB API: Allowing for live updates and enhanced metadata.

## 10. Conclusion

The Movie Recommendation System is an effective solution for helping users discover movies that match their tastes. By leveraging the TMDB 5000 Movies Dataset and combining content-based and collaborative filtering techniques, this project demonstrates the power of machine learning in creating personalized recommendation experiences.