🎬 Movie Recommendation System

# 📌 Project Title

Movie Recommendation System using Content-Based and Collaborative Filtering

# 🎯 Objective

To build a smart and interactive movie recommendation system that provides personalized movie suggestions to users using Content-Based Filtering and Collaborative Filtering techniques. The system includes a Streamlit GUI for seamless user interaction.

# 📂 Datasets Used

## 1. TMDB 5000 Movie Dataset

- Used for Content-Based Filtering  
- Contains movie metadata like genres, cast, crew, keywords, and overview  
- Helps generate similarity-based recommendations

## 2. MovieLens Small Latest Dataset

- Used for Collaborative Filtering  
- Includes user ratings for various movies  
- Useful for modeling user preferences based on interaction patterns

# 🧠 Recommendation Approaches

## 1. Content-Based Filtering

- Features Used: Genre, Director, Cast, Keywords  
- Technique: TF-IDF vectorization + Cosine Similarity  
- Goal: Recommend movies similar to a selected one based on metadata  
- Implementation: /Content based/ folder

## 2. Collaborative Filtering

- Approach: K-Nearest Neighbors (KNN) based Collaborative Filtering  
- Library Used: sklearn.neighbors.NearestNeighbors  
- Technique: Cosine distance between user vectors in the user-item matrix  
- Goal: Recommend movies liked by users with similar preferences  
- Implementation: /Collaborative Filtering/ folder

# 🛠️ Tech Stack

- Language: Python 3.x  
- Libraries: Pandas, NumPy, scikit-learn, Streamlit  
- Development Tool: Jupyter Notebooks  
- Interface: Streamlit GUI for interactive movie recommendations

# ▶️ How to Run

1. Clone the repository

git clone https://github.com/NandanPaT-eL/Movie-Recommendation-System.git  
cd Movie-Recommendation-System

2. Install dependencies

pip install -r requirements.txt

3. Launch the GUI

cd interface  
python main.py

# 🖥️ Interface Demo

- A simple Streamlit-based GUI allows users to select a movie and receive personalized recommendations.  
- Provides both content-based and collaborative suggestions.

# ⚔️ Challenges Faced

- Merging results from two different recommendation approaches  
- Handling sparse user-item matrices in collaborative filtering  
- Preprocessing and cleaning movie metadata from TMDB  
- Designing an intuitive yet simple GUI for user interaction

# 📚 Learnings

- Hands-on experience with recommendation algorithms (KNN, TF-IDF, Cosine Similarity)  
- Implementation of both content-based and collaborative recommender systems  
- Use of sklearn for building custom recommendation logic  
- GUI development with Streamlit for quick deployment and real-time testing