Movie Recommender System - Project Description

# 1. Project Overview

The Movie Recommender System is designed to help users discover films that match their preferences by analyzing a dataset of 5000 movies. The system employs machine learning techniques such as content-based filtering using cosine similarity and collaborative filtering to provide personalized recommendations.

# 2. Tools & Technologies

- Python

- Streamlit (for web app interface)

- NumPy

- Pandas

- Scikit-learn

- NLTK (for text preprocessing)

- Git & GitHub

# 3. Problem Statement

Users often face difficulty finding movies of interest within large databases due to extensive catalogs and subjective preferences. The goal is to build a system that provides relevant movie suggestions based on user preferences and past interactions.

# 4. Data Description

The dataset consists of information on 5000 movies, including attributes such as:  
- Title  
- Genres  
- Plot summaries or descriptions  
- Ratings (if available)  
- Other metadata (e.g., cast, director)

# 5. Approach & Methodology

5.1. Data Preprocessing

- Collected and cleaned data to handle missing values and inconsistencies.  
- Used NLTK for text preprocessing: tokenization, stop-word removal, and lemmatization on plot summaries.

5.2. Feature Engineering

- Extracted genre information and converted into vectors using techniques like one-hot encoding.  
- Generated text-based features (e.g., TF-IDF vectors) for plot descriptions.

5.3. Recommendation Techniques

- Content-Based Filtering: Calculated cosine similarity between movie feature vectors to recommend movies similar to a given movie or user preference.  
- Collaborative Filtering (if user rating data available): Used user-item interaction matrix to identify similar users or items and recommend accordingly.

5.4. Model Implementation

- Implemented algorithms using Scikit-learn for similarity computations.  
- Built a Streamlit interface to capture user input (e.g., favorite movies or genres) and display recommendations interactively.

# 6. Personal Contribution

- Designed and implemented data preprocessing pipelines in Python.  
- Developed feature extraction methods for genres and plot descriptions.  
- Implemented recommendation logic using cosine similarity and collaborative filtering.  
- Built the front-end interface using Streamlit for real-time user interactions.  
- Tested and validated recommendation results through user feedback and evaluation metrics.

# 7. Results & Outcomes

- Successfully generated relevant movie recommendations based on user preferences.  
- Received positive feedback from users on recommendation relevance.  
- Demonstrated the ability to scale the system to handle larger datasets.

# 8. How to Use

1. Clone the GitHub repository.  
2. Install required dependencies listed in requirements.txt (`pip install -r requirements.txt`).  
3. Run the Streamlit app: `streamlit run app.py`.  
4. Input favorite movie or select preferred genres to get recommendations.

# 9. Future Enhancements

- Integrate user authentication to collect user-specific ratings and improve collaborative filtering.  
- Expand dataset to include more movies and richer metadata (e.g., reviews, cast information).  
- Implement hybrid recommendation approaches combining content and collaborative methods.  
- Deploy the application on a cloud platform for wider accessibility.