Title: Movie Recommendation System

Abstract:

This project implements a movie recommendation system that suggests movies to users based on similarities in content. By analyzing movie metadata and using machine learning techniques, the system identifies relationships between different movies and generates relevant recommendations. This provides users with personalized suggestions, enhancing their viewing experience.

Introduction:

With the rapid growth of digital streaming platforms, recommendation systems have become an essential tool to improve user engagement. A movie recommendation system helps users discover movies they might enjoy without manually searching large collections. This project focuses on building a content-based recommendation system using Python, data preprocessing, and similarity measures.

Tools Used:

- **Python** – programming language for development
- **Pandas, NumPy** – data preprocessing and analysis
- **Scikit-learn** – machine learning utilities (vectorization, similarity)
- **NLTK** – natural language processing for text data
- **Streamlit** – interactive web application interface
- **Pickle** – model/data storage
- **TMDB API** – fetching movie posters

Steps Involved in Building the Project:

1. **Data Collection & Preprocessing**: Collected movie dataset and cleaned unnecessary fields.
2. **Feature Extraction:** Combined features like title, genre, keywords, cast, and director into a single text.
3. **Vectorization:** Applied CountVectorizer to transform text into numerical vectors.
4. **Similarity Computation**: Used cosine similarity to measure relationships between movies.

5. **Recommendation Function**: Designed a function that returns top similar movies for a given input.
6. **Web Application:** Built an interactive interface using Streamlit, allowing users to input a movie and view recommended movies with posters fetched via TMDB API.

Conclusion:

The project successfully demonstrates a working movie recommendation system that suggests relevant movies based on similarity. This can be further improved by integrating collaborative filtering, hybrid approaches, or deep learning methods to provide more accurate and personalized recommendations.