Movie Recommendation System

- Using Machine Learning and Streamlit

Introduction

Recommender systems have become vital in various industries such as e-commerce, entertainment, and content platforms. This project focuses on building a simple movie recommendation system using the MovieLens dataset. The goal is to suggest top movies to users based on their preferences by applying content-based filtering techniques and building a user-friendly interface using Streamlit.

Abstract

The Movie Recommendation System developed here is a machine learning-based project that helps users find movies they might like. It uses the genres of movies to compute similarity between them using TF-IDF vectorization and cosine similarity. Given a movie title, the system suggests five similar movies. A simple and interactive interface is built using Streamlit, allowing users to input their favorite movie and view personalized suggestions instantly.

Tools Used

- Python: Core programming language used for data processing and logic.
- Pandas: Used for data manipulation and reading CSV files.
- Scikit-learn: For vectorization (TF-IDF) and similarity computation.
- Streamlit: To build the web application interface.
- MovieLens Dataset: Open dataset containing movie metadata and ratings.

Steps Involved in Building the Project

1. Data Collection: Downloaded and extracted the movies.csv and ratings.csv from the MovieLens ml-latest-small dataset.

2. Data Preprocessing: Cleaned the movie genres data and filled missing values to prepare for analysis.

3. Feature Extraction: Applied TfidfVectorizer on the genre column to convert text into numerical vectors.

4. Similarity Calculation: Used cosine similarity to find movies similar to the input movie based on genre vector distance.

5. Recommendation Logic: Defined a function to retrieve top 5 similar movies using similarity scores.

6. Streamlit UI Development: Built a web interface using Streamlit allowing users to input a movie name and view recommendations.

7. Testing and Output: Ran the application locally, tested with various movie titles, and verified accurate recommendations.

Conclusion

This project successfully demonstrates how content-based filtering can be implemented for building a simple movie recommendation system. It highlights the power of machine learning in real-world applications and introduces the use of user-friendly tools like Streamlit to make the system interactive. The model can be further improved by integrating collaborative filtering, sentiment analysis, or deep learning for better personalization.