CODSOFT – Task: Movie Recommendation System (Web App)

Abstract

This project is part of the CODSOFT Artificial Intelligence Internship. It is a content-based movie recommender system built using Python, Flask, Pandas, and Scikit-learn. The system suggests movies similar to the user's input based on title and genres. It uses the MovieLens dataset for training and powers a modern web application with an attractive UI.

Introduction

Recommendation systems are widely used in modern platforms such as Netflix, Amazon, and YouTube to suggest items similar to a user's preference. In this project, we build a simple content-based movie recommender using TF-IDF and cosine similarity. The project also includes a web interface powered by Flask and styled with modern CSS.

Dataset

We use the MovieLens (ml-latest-small) dataset containing thousands of movies with metadata such as movieId, title, and genres. This dataset is suitable for building and testing recommender systems.

Methodology

1. Load dataset (movies.csv).
2. Preprocess by combining title and genres into a text field.
3. Use TF-IDF Vectorizer to convert text into numerical vectors.
4. Apply cosine similarity to find the most similar movies.
5. Build a Flask web application to serve recommendations.
6. Style the frontend with HTML, CSS, and JavaScript.

Implementation

The project is structured into multiple folders for clarity:
- app/: contains recommender_core.py and webapp.py
- data/: contains movies.csv dataset
- static/: contains style.css and optional JSON files
- templates/: contains index.html for frontend UI
- venv/: Python virtual environment

- setup_and_run.sh: setup and execution script

The user interacts with the web application via a search bar. Upon typing a movie name, the system fetches similar movies and displays them with genres in a neat results panel.

Screenshots (Sample)

[Add screenshots of homepage, search results, and recommendations here.]

Conclusion

This project demonstrates how recommendation systems can be built using simple machine learning techniques. It also provides hands-on experience with Flask web development, frontend design, and data preprocessing. Future improvements include adding movie posters using TMDB API and hybrid recommendation methods.

References

- MovieLens dataset: https://grouplens.org/datasets/movielens/
- Flask documentation: https://flask.palletsprojects.com/
- Scikit-learn documentation: https://scikit-learn.org/