Movie Recommender System – Project Report

Submitted by: Gaurav Waghmare

Date: July 2025

Introduction

With the explosive growth of digital content, choosing a movie to watch has become increasingly challenging. Personalized movie recommender systems help users navigate this vast landscape by filtering options based on individual preferences. This project presents an interactive web application that enables users to discover movies based on genres and years, enhanced with posters and real-time ratings.

Abstract

The Movie Recommender System is designed to provide viewers with an optimized browsing experience. Users select genres and release years, receiving tailored movie recommendations, each enriched with poster images and aggregated ratings from global sources such as IMDb, Rotten Tomatoes, and Metacritic via the OMDb API. Built using Streamlit and Python, the application delivers rapid, visually appealing, and informative results suitable for all audiences.

Tools Used

Tool	Purpose
Python	Core programming language
Pandas	Data processing and filtering
Streamlit	Interactive user interface
Requests	API interactions (OMDb)
OMDb API	Movie metadata and poster retrieval
CSV Dataset	Source for movie titles and genres

Steps Involved in Building the Project

1. **Data Preparation**
    - Loaded and preprocessed movies.csv to obtain movieId, title, and genres.
    - Extracted the release year from movie titles using regular expressions.
2. **User Interface with Streamlit**
    - Implemented sidebar filters for genre and year selection.
    - Displayed unique, selectable genre options.

3. **API Integration**
    - Queried the OMDb API for each filtered movie to obtain real-time poster images and ratings from multiple sources.
    - Used fallbacks for posters or ratings if unavailable.
4. **Visual Display**
    - Rendered movie information as visually engaging cards using HTML and CSS within Streamlit's markdown.
    - Included the movie poster, title, genres, year, and rating details.
5. **Performance Optimization**
    - Employed Streamlit’s caching to minimize redundant API calls and improve app responsiveness.
    - Applied error handling for missing or incorrect data, ensuring a smooth user experience.

Conclusion

The Movie Recommender System demonstrates how combining a basic dataset, real-time API integration, and modern web frameworks can create an engaging recommendation platform. The app is scalable and can be extended to incorporate advanced machine learning-based recommendations, user login support, or real-time collaborative filtering. This foundation offers a practical showcase of the synergy between data science, web development, and cloud APIs in delivering meaningful user experiences.