

Movie Recommendation System (Content-Based Filtering): -

1 Introduction

In the age of digital streaming platforms, movie recommendation systems have become crucial in enhancing user experience by helping viewers discover content that matches their preferences. This project focuses on developing a **content-based movie recommendation system** using genre metadata from movies. The system analyses the genres associated with a movie and suggests other movies with similar genre profiles.

2 Abstract

This project implements a movie recommender system using content-based filtering techniques. We use the **The Movies Dataset** (sourced from Kaggle) which contains metadata for approximately 45,000 movies, including title and genre information. The recommendation system utilizes text feature extraction and cosine similarity to compute the closeness of movies based on their genre compositions.

Due to memory constraints, we worked with a subset of 1000 movies for efficient similarity computation. The system suggests the top 5 most similar movies based on genre similarity for a given input movie title. This project highlights the effectiveness of simple content-based techniques in providing personalized movie suggestions without requiring user ratings.

3 Dataset

- **Name:** The Movies Dataset
 - **Source:** Kaggle (<https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset>)
 - **File Used:** movies_metadata.csv
 - **Features:**
 - title — Movie title
 - genres — List of genres for the movie
-

4 Tools & Technologies Used

- **Python 3.x**
- **Pandas:** Data manipulation
- **Scikit-learn:** CountVectorizer, cosine similarity
- **Jupyter Notebook:** Development environment

5 Steps Involved in Building the Project

✓ Step 1: Data Loading and Cleaning

- Loaded movies_metadata.csv using Pandas.
- Selected title and genres columns.
- Parsed the JSON-like genres field to extract plain genre names.
- Removed duplicate titles and rows with empty genres.

✓ Step 2: Vectorization of Genres

- Used CountVectorizer to convert genre text data into numerical vectors where each feature represents the presence of a genre.

✓ Step 3: Similarity Computation

- Computed cosine similarity scores between movie vectors to measure genre similarity.
- Limited data to 1000 movies to avoid memory issues when building the similarity matrix.

✓ Step 4: Recommendation Logic

- For a given input movie, retrieved its vector.
- Found top 5 most similar movies (excluding itself) based on cosine similarity scores.

6 Conclusion

This project demonstrates how content-based filtering can be applied to movie recommendation using just metadata, without requiring user interaction history. The system is memory-efficient and provides meaningful suggestions based on genre similarity.

Future improvements could include combining content-based methods with collaborative filtering for a hybrid system, integrating additional features like keywords and crew information, and developing a user-friendly web interface using Streamlit.