

ELEVATE LAB INTERNSHIP PROJECT REPORT

Project Title: Movie Recommendation System

Intern Name: PUNAM KOLI

Duration: 24 April 2025 to 23 May 2025

1. Introduction

In today's digital world, users are overwhelmed by the sheer volume of content available online. Recommender systems offer a smart way to filter and personalize content, enhancing user satisfaction. This project focuses on developing a content-based Movie Recommendation System that suggests similar movies to users based on the textual metadata of movies such as genres, overview, cast, and crew.

2. Abstract

The objective of this project was to create an intelligent movie recommendation engine using machine learning and content filtering techniques. A dataset from **Kaggle** was used, containing comprehensive metadata for popular movies. Using **text vectorization** and **cosine similarity**, the model identifies and recommends the top 5 most similar movies to any selected movie. The backend logic was implemented in **Jupyter Notebook**, and the final system was deployed as a fully interactive web application using **Streamlit** in **PyCharm**. To optimize performance, serialized files (`movie_dict.pkl` and `similarity.pkl`) were used to store preprocessed data and similarity metrics.

3. Tools & Technologies Used

- **Languages:** Python
- **Libraries:** Pandas, NumPy, Scikit-learn, Pickle, Requests
- **IDE & Platform:** Jupyter Notebook, PyCharm
- **Web Framework:** Streamlit
- **API Integration:** OMDb API (for movie posters)
- **Dataset Source:** Kaggle

4. Dataset -<https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata>

5. Steps Involved in Building the Project

a. Data Preprocessing

- Loaded and cleaned the dataset (handling missing values, formatting issues)
- Merged textual features (overview, genres, keywords, cast, crew)
- Removed duplicates and null entries to improve data quality

b. Text Vectorization

- Used `CountVectorizer` to convert combined text features into numerical vectors
- Generated token count vectors with a max feature limit to optimize performance

c. Model Building

- Computed cosine similarity matrix using vectorized data
- Stored the processed data and similarity matrix into `movie_dict.pkl` and `similarity.pkl` for efficient use

d. Website Integration (Streamlit)

- Designed an interactive web app with a search dropdown for movie selection
- Implemented a **recommendation function** to fetch and display similar movie titles
- Used **OMDb API** to fetch real-time movie posters
- Styled the web interface using custom CSS for a modern, clean look

6. Conclusion

This project effectively demonstrates how content-based filtering can be used to build a practical recommendation engine. From data preprocessing to building a machine learning model and deploying a user-friendly web application, this experience deepened my understanding of real-world AI applications. The modular structure and seamless integration make it scalable and ready for further enhancements like hybrid recommendations or user-based filtering.

7. Github Repository-

https://github.com/Punamkoli14/Project_AL_ML_MovieRecommendationSystem

8. Output-

https://github.com/Punamkoli14/Project_AL_ML_MovieRecommendationSystem/blob/main/Screenshot2-MovieRecommendationSystem.png