

MOVIE RECOMMENDATION SYSTEM USING PYTHON

PROJECT REPORT

by

LAKSHYA KALITA

Roll Number :- 28840524001

SOHAN MALLICK

Roll Number :- 28840524013

Department of BSc in Data Science
Institute of Advance Education & Research
Kolkata
May, 2025

BONAFIDE CERTIFICATE

This is to certify that this project report entitled “**MOVIE RECOMMENDATION SYSTEM USING PYTHON**” submitted to **Institute of Advance Education & Research, Kolkata**, is a bonafide record of work done by “**LAKSHYA KALITA & SOHAN MALLICK**” under my supervision from “ / / ” to “ / / ”

Somnath Koley

Head of the Department

Department of BSc in Data Science

Place

Date

Declaration by Authors

This is to declare that this report has been written by us. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be plagiarized, we shall take full responsibility for it.

Lakshya Kalita

Roll number: 28840524001

Sohan Mallick

Roll number: 28840524013

Place

Date

Abstract

This report presents a movie recommendation system built using Python, leveraging content-based filtering and certain amount of popularity based filtering. The system utilizes datasets like Kaggle for movie metadata and user ratings. Key techniques include cosine similarity for finding similar movies and matrix vectorization for user preferences. Implementation involves data preprocessing, feature extraction, and model training using libraries like Pandas, NumPy, Scikit-learn, and DiffLib. The project successfully provides personalized movie recommendations based on content and popularity with scope for deep learning and hybrid models in future improvements.

TABLE OF CONTENTS

CHAPTER NO. TITLE	PAGE NO.
ABSTRACT	iii
1. Introduction.....	1
1.1 Overview of Recommendation System.....	1
1.2 Importance of Movie Recommendation Systems.....	2
1.3 Objective of the Project.....	2
2. Technology & Methodology.....	3
2.1 Tools & Libraries Used.....	3
2.2 Data Collection & Preprocessing.....	3
2.3 Feature Extraction.....	3
2.4 User Input.....	3
2.5 Cosine Similarity.....	3
2.6 Algorithms Implemented.....	3
3. Implementation.....	4
4. Conclusion & Future Work.....	5
4.1 Summary.....	5
4.2 Future Improvements.....	5
4.3 Real-World Applications.....	5
5. REFERENCES.....	6

1. Introduction

1.1 Overview of Recommendation Systems

Recommendation systems are algorithms designed to predict user preferences and suggest relevant items. They are widely used in e-commerce, music streaming, and online movie platforms like Netflix, Amazon prime and IMDb

There are three primary types of recommendation systems:

i. Content-Based Filtering

- Recommends movies similar to what a user has previously liked, based on attributes like genre, director, and description.
- Uses cosine similarity or TF-IDF vectorization to measure similarity between movie features.
- Example: If a user enjoys sci-fi movies, it recommends similar sci-fi titles.

ii. Popularity-Based Filtering

- Recommends movies based on their overall popularity, considering factors like average rating and total number of reviews.
- Works well for general recommendations but lacks personalization.
- Example: Suggesting top-rated or most-watched movies globally.

iii. Collaborative Filtering

- Predicts user preferences by analyzing past interactions and similarities between users.
- Two types: User-based (finds users with similar tastes) and Item-based (finds movies liked by similar users).
- Example: If User A and User B have similar ratings for many movies, the system recommends movies User A liked to User B.

These techniques help improve user experience by providing personalized suggestions, increasing engagement, and enhancing content discovery.

1.2 Importance of Movie Recommendation Systems

With the massive number of movies available, users often struggle to find content that matches their preferences. A movie recommendation system helps by analysing user preferences and movie features to provide personalized suggestions. They are widely used by streaming platforms like Netflix, Amazon Prime, and Disney+, helping improve customer satisfaction and retention.

1.3 Objective of the Project

The goal of this project is to develop a movie recommendation system using Python, leveraging content-based filtering, collaborative filtering, and popularity-based filtering. The system analyses user interactions, movie metadata, and similarity metrics to provide personalized recommendations. By implementing techniques like cosine similarity, matrix vectorization, and feature extraction, this project aims to enhance user experience and improve recommendation accuracy.

2. Technology & Methodology

2.1 Tools & Libraries Used

- Python (Programming language)
- Pandas & NumPy (Data manipulation)
- Scikit-learn (Machine learning)
- DiffLib (provides classes and functions for comparing sequences)

2.2 Data Collection & Preprocessing

- **Dataset:** Using Kaggle datasets.
- **Cleaning:** Handling missing values, duplicates, and irrelevant data.
- **Feature Engineering:** Extracting metadata such as actors, genres, and director information.

2.3 Feature Extraction

Feature extraction helps improve the recommendation quality. Common extracted features include:

- **Genre Information** (Categorical encoding)
- **Movie Descriptions** (Text-based similarity using NLP)
- **User Ratings** (Numerical data for collaborative filtering)

2.4 User Input

- Users can search for movies titles, or specify their preferred genres.
- The system adapts recommendations based on user interactions.

2.5 Cosine Similarity

Used in content-based filtering, cosine similarity measures how similar two movies are based on their feature vectors. This helps find movies that are most similar to a given input.

2.6 Algorithms Implemented

Content-Based Filtering: Recommends movies similar to previously liked ones.

Popularity-Based Filtering: Recommends movies based on their overall popularity.

3.Implementation

The implementation of the movie recommendation system follows three main approaches: data loading & pre processing, content-based filtering, and popularity-based filtering.

- **Data Loading & Preprocessing** – The dataset (e.g., Kaggle) is cleaned, missing values are handled, and relevant features like genre, ratings, and descriptions are extracted.
- **Content-Based Filtering** – Uses TF-IDF vectorization and cosine similarity to find movies similar to a given movie based on descriptions or metadata.
- **Popularity-Based Filtering** – Recommends top-rated movies by sorting based on average rating and number of reviews, ensuring widely liked content is prioritized.

These methods are combined to enhance recommendation accuracy, offering users personalized and relevant movie suggestions.

4. Conclusion & Future Work

4.1 Summary

This project successfully implemented a movie recommendation system using cosine similarity, popularity filtering, and content-based filtering.

4.2 Future Improvements

1. **Hybrid Models:** Combine content-based and collaborative filtering to improve accuracy.
2. **Deep Learning:** Use neural networks for better feature extraction.
3. **User Feedback Loop:** Continuously refine recommendations based on new ratings.

4.3 Real-World Applications

- **Streaming Services:** Netflix, Disney+, Prime Video.
- **E-Commerce:** Amazon product recommendations.
- **Music Platforms:** Spotify playlist recommendations.

This report provides a detailed overview, including concept explanations, and analysis.

REFERENCES

Here are some useful references for building a movie recommendation system using Python:

1. [TechVidvan Guide](#) – A detailed tutorial on implementing a movie recommendation system using machine learning techniques.
2. [CodeWithFaraz](#) – A step-by-step guide covering data preprocessing, feature extraction, and similarity computation for movie recommendations.

These sources provide insights into data handling, algorithms, and implementation techniques.