NAME: Dinesh karthik nayak

MAIL ID: dineshnayak2436@gmail.com

Movie Recommendation System Project Report

---

1. Introduction: In an age where streaming platforms offer thousands of movies, viewers often struggle with the question: "What should I watch next?" The Movie Recommendation System is designed to solve this problem. By understanding users' past preferences and analyzing movie features, the system intelligently recommends films tailored to individual tastes. This project is a practical implementation of machine learning in everyday life, aiming to make content discovery seamless and enjoyable.

---

2. Abstract: This project involves designing a movie recommendation system that leverages the MovieLens 100K dataset to provide personalized suggestions. It combines two popular recommendation strategies: collaborative filtering (which learns from user behavior) and content-based filtering (which focuses on movie genres). The system is implemented in Python and brought to life through an intuitive web interface using Streamlit. Users can select any movie and receive five personalized suggestions, enhancing the decision-making process and providing a more engaging viewing experience.

---

3. Tools Used:

- **Python**: The backbone of the project, used for building models, data processing, and logic implementation.
- **Pandas**: Used extensively for loading, cleaning, and manipulating structured data.
- **Scikit-learn**: Applied for measuring similarities between users and movies using cosine similarity.
- **Streamlit**: A lightweight framework used to create an easy-to-use and interactive web interface (UI).
- **MovieLens 100K Dataset**: A real-world dataset containing 100,000 ratings from 943 users on 1,682 movies.

---

4. Steps Involved in Building the Project:

1. **Data Collection:** Downloaded the MovieLens 100K dataset, a widely used benchmark dataset for movie recommendation tasks. It contains user ratings for a variety of movies along with movie metadata like titles and genres.
2. **Data Preprocessing:** Cleaned the dataset by removing missing values and merged the user ratings with movie metadata. The genres were processed into binary vectors (multi-hot encoding) to represent the presence or absence of each genre.
3. **Collaborative Filtering:** Created a user-item interaction matrix where each cell represented the rating a user gave to a movie. Cosine similarity was used to compute the similarity between items. Based on a selected movie, similar movies were recommended by finding the highest similarity scores.
4. **Content-Based Filtering:** Movies were represented using genre-based vectors. Cosine similarity was again applied to compare these vectors, allowing the recommendation of movies with similar content or genre profiles.
5. **Interface Design:** Streamlit was employed to develop an interactive and user-friendly UI. Users could select a movie and filtering method from dropdown menus. Once selections were made, the system instantly displayed top 5 movie recommendations.
6. **Recommendation Logic:** The recommendation engine processed the user's input and dynamically returned similar movie suggestions based on the chosen strategy (content-based or collaborative). The interface refreshed automatically to reflect changes.
7. **Testing and Validation:** Several test inputs were evaluated to verify the relevance of recommendations. Results consistently showed meaningful movie suggestions, confirming that the filtering logic was effective.

---

5. Conclusion: The Movie Recommendation System successfully combines collaborative and content-based filtering to deliver relevant movie suggestions to users. Its interactive interface makes it accessible for general audiences, and its modular design allows future improvements. Enhancements like integrating sentiment analysis, personalized user profiles, and cloud deployment could significantly increase its impact. Overall, this project demonstrates how machine learning can enrich digital experiences by making content discovery more intelligent and efficient.