REPORT FILE

Introduction:

Recommender systems are essential in modern applications like Netflix, Amazon, and YouTube. They help users discover content based on past behavior and preferences. This project explores popular recommendation techniques using a dataset of movies and user ratings.

Abstract:

This project aims to build a movie recommendation system that suggests films to users based on their preferences. We employ collaborative filtering and content-based filtering methods, leveraging user ratings and movie metadata.

Tools used:

1. Programming Language

- **Python** – Primary language for data analysis, modeling, and visualization

2. Data Manipulation & Analysis

- **Pandas** – For data loading, preprocessing, and manipulation
- **NumPy** – For numerical computations

3. Data Visualization

- **Matplotlib** – For plotting charts and graphs
- **Seaborn** – For statistical data visualization

4. Machine Learning Libraries

- **Scikit-learn** – For building content-based filtering models (e.g., TF-IDF, cosine similarity)

5. Natural Language Processing (NLP)

- **NLTK** – For text-based feature extraction from genres, descriptions, etc.

6. Development Environment

- **Jupyter Notebook** – For interactive coding, visualizations, and documentation

7. Version Control

- **Git** – For code version control

- **GitHub** – To host project code and collaborate

8. Deployment

- **Streamlit** – To create a simple web app interface for the recommender system
- **Heroku** – To deploy the app online

Steps involved:

1. Problem Definition

- Understand the objective: Build a system that recommends movies to users based on their past preferences or similar movies.

2. Data Collection

- Collected data from the Kaggle
- Loaded data files .

3. Data Preprocessing

- Merged and cleaned datasets (e.g., combining movie titles with ratings)
- Handled missing values and duplicates
- Extracted features (e.g., year from title, multi-hot encoding of genres)
- Converted data types where necessary

---

4. Exploratory Data Analysis (EDA)

- Analyzed data distributions (e.g., rating frequency, popular genres)
- Visualized most watched and top-rated movies
- Identified trends in user behavior

5. Model Building

A. Content-Based Filtering

- Recommended similar movies using genre-based similarity
- Used TF-IDF + cosine similarity for similarity scoring

B. Collaborative Filtering

- Applied memory-based.
- Implemented model-based methods using matrix factorization .

C. Hybrid Recommendation (if implemented)

- Combined both content and collaborative scores to recommend movies

7. Deployment (Optional)

- Built a basic web application using **Streamlit** .

8. Documentation and Reporting

- Compiled findings, code, visualizations, and results into a formal project report
- Documented challenges, insights, and future work

Conclusion:

The system successfully recommends personalized movies using both user behavior and movie content. Future improvements may include using deep learning models, contextual data (e.g., time of day), or reinforcement learning techniques.