Sri Lakshmi Sonti
801401418

Movie Recommendation System Report

1. Introduction

Recommendation systems play a major role in modern digital platforms, helping users discover relevant content across movies, music, shopping, and social media. In this project, we build a movie recommendation system using the Movie-Lens 100K dataset.

We explore three approaches:

1. User-based collaborative filtering, which recommends movies by finding users with similar taste.

2. Item-based collaborative filtering, which recommends movies similar to a given movie.

3. Pixie-inspired random walk recommendation, a graph-based approach that simulates how users move through content networks in real-world platforms.

These methods demonstrate different strengths and weaknesses and together give a complete understanding of how recommendation systems work in practice.

2. Dataset Description

The project uses the Movie-Lens 100K dataset, which contains:

- 943 users
- 1,682 movies
- 100,000 ratings

Datasets used include:

- u.data — user ratings (user_id, movie_id, rating, timestamp)
- u.item — movie metadata (movie_id, title, release_date, etc.)
- u.user — user demographic information

Preprocessing steps included:

- Loading data using the correct separators (|, \t)
- Handling missing values
- Converting timestamps to readable dates
- Creating cleaned CSVs: ratings.csv, movies.csv, and users.csv

These cleaned datasets were used in all subsequent tasks.

3. Methodology

We implemented three recommendation techniques:

3.1 User-Based Collaborative Filtering

- Constructed a user-movie matrix using pivot().
- Filled missing values with 0 for similarity computation.
- Calculated cosine similarity between users.
- Recommended movies based on the weighted average ratings from the most similar users.

3.2 Item-Based Collaborative Filtering

- Transposed the user-movie matrix to compute similarity between movies.
- Computed cosine similarity for item-item pairs.
- Recommended movies that have the highest similarity to a given movie title.

- Also implemented a title-based similarity approach using CountVectorizer.

3.3 Pixie-Inspired Graph-Based Recommendation

- Created a bipartite graph connecting users to movies they rated.
- Performed random walks starting from a movie or movie ID.
- Counted the number of visits to each movie.
- Ranked the most-visited movies as recommendations.

This approach mimics real-time graph traversal used in large-scale systems.

4. Implementation Details

- The dataset was merged using movie_id to attach titles to ratings.
- Ratings were mean centered per user to remove personal rating bias.
- Two adjacency lists were created:
    - user_to_movies[user] → set of movies
    - movie_to_users[movie] → set of users
- The random walk algorithm:
    - Alternates between movie → user → movie.
    - Runs 100 walks of length 15.
    - Stores visit frequencies using a Counter.
- The Pixie implementation supports:
    - Movie title input
    - Movie ID input (numeric)

Top-N results are returned as a DataFrame for easy viewing.

5. Results and Evaluation

Each model produced meaningful recommendations such as:

User-Based Filtering (for user 10)

- In the Company of Men
- Les Misérables
- Thin Blue Line
- Braindead
- Boys

Item-Based Filtering ("Jurassic Park (1993)")

- Top Gun
- Empire Strikes Back
- Raiders of the Lost Ark
- Indiana Jones and the Last Crusade
- Speed

Random Walk (Pixie-Based)

“Jurassic Park (1993)” produced movies frequently co-watched by users:

- Mighty Aphrodite
- Twelve Monkeys
- Mission Impossible
- Leaving Las Vegas
- My Own Private Idaho

Observations:

- User-based filtering works well when a user has many ratings.
- Item-based filtering performs best with strong item similarity signals.
- Pixie-based recommendations are diverse and organic, capturing hidden relationships.

6. Conclusion

This project demonstrates how different recommendation strategies can be applied to the same dataset. User-based and item-based collaborative filtering provide structured, similarity-based recommendations, while the Pixie-style random walk approach captures deeper graph relationships and yields more exploratory recommendations.
Future improvements include:

- Hybrid recommendation models
- Adding genre, year, and metadata features
- Using embedding-based deep learning models

These enhancements can bring the recommendation system closer to real-world commercial systems.