Title: Movie Recommendation System Report

1. Introduction

Movie recommendation systems are crucial in today's digital landscape, helping users discover content they might enjoy. This notebook explores three different approaches to movie recommendation: user-based collaborative filtering, item-based collaborative filtering, and a graph-based Pixie-inspired algorithm.

2. Dataset Description

The MovieLens 100K dataset is used, which contains 100,000 ratings from 943 users on 1682 movies. The dataset includes user IDs, movie IDs, ratings, timestamps, movie titles, and user demographics. Preprocessing steps involve loading the data, handling missing values, and normalizing ratings.

3. Methodology

- **User-based collaborative filtering:** This approach recommends movies to a user based on the preferences of similar users. User similarity is calculated using cosine similarity on the user-movie rating matrix.
- **Item-based collaborative filtering:** This approach recommends movies that are similar to movies a user has liked. Item similarity is calculated using cosine similarity on the transposed user-movie rating matrix.
- **Random-walk-based Pixie algorithm:** This approach represents user-movie interactions as a graph and uses random walks to recommend movies. The algorithm starts at a user or movie node and randomly walks through the graph, recommending movies that are frequently visited during the walk.

4. Implementation Details

- **Data Loading and Cleaning:** The notebook loads the u.data, u.item, and u.user datasets using pandas. It handles missing values by filling them with 0 or the mean rating. It also converts timestamps to readable dates.
- **User-Based Collaborative Filtering:** The recommend_movies_for_user() function computes user similarity using cosine similarity and recommends movies based on the weighted average of ratings from similar users.
- **Item-Based Collaborative Filtering:** The recommend_movies() function computes item similarity using cosine similarity and recommends movies based on the similarity scores.
- **Graph-Based Pixie Algorithm:** The notebook constructs an adjacency list graph where users and movies are nodes, and edges represent ratings. The weighted_pixie_recommend() function performs random walks on the graph and recommends movies based on visit frequency.

5. Results and Evaluation

The notebook provides example outputs from each recommendation approach. User-based collaborative filtering recommends movies based on similar users' preferences. Item-based collaborative filtering recommends movies similar to those a user has liked. The Pixie algorithm explores the user-movie graph to find relevant recommendations.

Limitations include the cold-start problem (recommending movies to new users or recommending new movies) and the reliance on explicit ratings.

6. Conclusion

This project demonstrates three different approaches to movie recommendation. Potential improvements include using hybrid models that combine different approaches, incorporating additional features such as movie genres and user demographics, and using more sophisticated graph-based algorithms. These methods have real-world applications in various industries, such as e-commerce, social media, and content streaming.