Shahid Beheshti University
Machine Learning
M.Sc. - Fall 2024

Final Project: Movie Recommender Systems

Project Overview

This project focuses on building a movie recommender system using a rich dataset of movie metadata and user ratings. By working on this task, you will explore various recommendation algorithms, analyze their strengths and weaknesses, and implement one or more methods to generate meaningful recommendations.

Objectives

Your task is to develop a movie recommender system that suggests movies to users based on their preferences and viewing history. This project is designed to:

1. Introduce you to recommendation systems.
2. Explore multiple recommendation algorithms and evaluate their effectiveness.
3. Gain experience working with large datasets and implementing data-driven solutions.

Dataset Description

The dataset contains metadata and user ratings for 45,000 movies. Here are the key files included:

- **movies_metadata.csv:** Information about movies, including titles, release dates, budget, revenue, production details, and more.
- **keywords.csv:** Contains plot keywords for the movies in JSON format.
- **credits.csv:** Details about the cast and crew of the movies in JSON format.

- **links.csv:** TMDB and IMDB IDs for all movies in the dataset.
- **links_small.csv:** A smaller subset with TMDB and IMDB IDs for 9,000 movies.
- **ratings_small.csv:** A subset of 100,000 ratings provided by 700 users for 9,000 movies. Ratings are on a scale of 1-5.

Tasks and Instructions

1. Data Exploration and Preprocessing

- Analyze the provided dataset to understand its structure and identify useful features.
- Handle missing or inconsistent data.
- Process JSON columns (e.g., keywords, cast, crew) for meaningful insights.
- Normalize and preprocess data as required for your chosen algorithms.

2. Explore Recommendation Algorithms

You are required to research and explore the following types of recommendation systems:

1. **Popularity-Based Filtering:**
   - Recommends movies based on their overall popularity (e.g., average rating, number of votes).
   - Simple and effective for new users with no prior interaction history.
2. **Content-Based Filtering:**
   - Recommends movies similar to what a user has liked in the past.
   - Use movie metadata (e.g., genres, keywords, cast, crew) to build feature vectors.
3. **Collaborative Filtering:**
   - Recommends movies based on user ratings.
   - Two main approaches:
     - User-based: Finds users with similar preferences.
     - Item-based: Finds items (movies) rated similarly by users.
4. **Hybrid Systems:**
   - Combine content-based and collaborative filtering for improved performance.
5. **Advanced Techniques (Optional):**
   - Matrix factorization (e.g., SVD, NMF)
   - Deep learning approaches (e.g., autoencoders for collaborative filtering)

3. Implement Your Recommender System

- Select one or more algorithms from the explored options.
- Justify your choice based on the dataset and project requirements.
- Implement your chosen algorithm(s) and generate recommendations.

4. Evaluate Your Recommender System

- Use appropriate metrics to evaluate the performance of your recommender system.
- Compare the performance of different algorithms if implementing more than one.

Project Deliverables

- **Code Implementation:**
  - Reproducible code with detailed documentation.
  - Include an inference pipeline to demonstrate recommendations.
- **Report:**
  - Dataset analysis and preprocessing steps.
  - Explanation of the algorithms explored.
  - Justification for your chosen algorithm(s).
  - Evaluation results and analysis.
  - Insights and recommendations for improving the system.

Keywords to Research

- Popularity-based filtering
- Content-based filtering
- Collaborative filtering
- Matrix factorization
- Hybrid recommendation systems
- Evaluation metrics for recommender systems

Additional Notes

- Be creative in feature engineering and data preprocessing.
- Leverage libraries such as surprise, lightfm, or Hugging Face's pretrained models for advanced techniques.
- Think critically about the scalability and usability of your system.

We encourage you to experiment, analyze, and present a well-rounded solution. Best of luck with your project!