Movie Recommendation System

Personalizing Entertainment through Data-Driven Insights

Overview

With an ever-expanding catalog of movies on digital platforms, users face the challenge of selecting content that matches their tastes. Recommendation systems play a critical role in enhancing user experience by suggesting relevant movies based on individual preferences. This project explores the development of a hybrid movie recommendation engine using the MovieLens 100k dataset, integrating collaborative filtering and content-based methods.

Objective

To design and implement a recommendation system that:

- Suggests movies tailored to individual user preferences.
- Increases user engagement by reducing decision fatigue.
- Enhances platform retention through accurate and enjoyable recommendations.

Business Problem

Streaming platforms often encounter the following challenges:

- Users are overwhelmed by choice, leading to poor engagement.
- Lack of personalization can increase churn and reduce user satisfaction.
- Recommending popular content may ignore niche preferences, missing opportunities to engage diverse audiences.

An effective recommendation system addresses these challenges by learning from user behavior and content characteristics to provide customized movie suggestions.

Dataset Description

The **MovieLens 100k** dataset, curated by the GroupLens research group, contains 100,000 ratings (1–5) from 943 users on 1,682 movies. It includes the following components:

- **User Information:**
    - User ID
    - Age

- Gender
- Occupation
- Zip-code

- **Movie Information:**
  - Movie ID
  - Title
  - Genre(s)
  - Release Date
- **Ratings:**
  - User ID
  - Movie ID
  - Rating (1–5)
  - Timestamp

Data Source:
[MovieLens 100k Dataset – Kaggle]

Proposed Solution: Methodology

1. Data Preparation and Exploration

- Load and merge datasets (movies, users, and ratings).
- Clean data to handle missing values and inconsistent genres.
- Perform exploratory data analysis (EDA) to understand user behavior and rating patterns.

2. Feature Engineering

- Extract features such as release decade, genre flags, and user demographics.
- Encode categorical variables like genre and occupation.
- Create user-item interaction matrices.

3. Model Development

- Implement and compare two approaches:

- **Collaborative Filtering** (User-Based & Item-Based KNN, Matrix Factorization using SVD)
- **Content-Based Filtering** (using genres and metadata)

4. Model Evaluation

- Use RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) for rating prediction accuracy.
- Evaluate ranking quality using metrics such as Precision@K and Recall@K.

5. Insight Generation

- Analyze genre preferences by demographic groups.
- Identify top-rated and most-watched movies across segments.
- Use clustering (e.g., K-Means) to identify user segments.

Deliverables

- **Recommendation Engine:** Trained hybrid model combining collaborative and content-based filtering.
- **Insights Report:** Trends in viewing behavior, influential user features, and genre preferences.
- **Codebase:** Python scripts with documentation and reproducible Jupyter notebooks.
- **Presentation Deck:** Visual storytelling of methodology, findings, and recommendations.

Key Insights

- **Hidden Preferences:** Matrix factorization reveals latent preferences not explicit in user profiles.
- **Genre Influence:** Genre-based features significantly improve content-based recommendations.
- **User Segmentation:** Young users tend to rate more action and sci-fi movies higher, while older audiences prefer drama and documentary genres.
- **Cold-Start Solution:** Content-based filtering can support new users or movies lacking historical interaction data.

Constraints and Assumptions

- **Data Limitations:** Ratings are explicit; implicit behaviors (like watch time or skips) are not captured.
- **Cold-Start Problem:** New users or movies with no interaction history can affect collaborative filtering performance.
- **Bias and Fairness:** Recommendations may reflect historical popularity biases unless adjusted.

Next Steps

- **Model Tuning:** Optimize hybrid models using advanced techniques like neural collaborative filtering or LightFM.
- **Deployment:** Create a user-friendly interface to input preferences and receive recommendations.
- **Scalability:** Expand to larger datasets (e.g., MovieLens 1M or Netflix Prize) for real-world relevance.
- **A/B Testing:** Evaluate real-world effectiveness by testing on a user cohort.

Conclusion

By leveraging both collaborative and content-based approaches, this project builds a robust movie recommendation system aimed at enhancing user experience. Personalization not only improves engagement and satisfaction but also provides streaming platforms with a competitive edge in retaining users and promoting content effectively.