

Rsumaili / Phase-4-Project

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

☆ 0 stars

🔗 0 forks

👁 0 watching

🔗 Branches

📈 Activity

🏷 Tags

🌐 Public repository

1 Branch

0 Tags

Go to file

t

Go to file

+

Add file

Code

...

dantonki updated nontechnical presentation

a70da8b · 2 minutes ago 🕒

📁 .ipynb_checkpoints	updated nontechnical presentation	2 minutes ago
📁 Images	EDA graphs and recommendation su...	5 minutes ago
📁 ml-latest-small/ml-latest-small	Added notebook and data used in t...	yesterday
📄 Non-Technical Presentation.pdf	updated nontechnical presentation	2 minutes ago
📄 Non-Technical Presentation.p...	updated nontechnical presentation	2 minutes ago
📄 README.md	Added some content to the file	1 hour ago
📄 index.ipynb	Detailed notebook	1 hour ago

📖 README

Movie Recommendation System

This project demonstrates a personalized movie recommendation system using the [MovieLens 100K dataset](#). It leverages collaborative filtering and hybrid modeling techniques to recommend movies based on user preferences.

Project Overview

- **Goal:** Build a movie recommender system that provides accurate and relevant movie suggestions to users.
- **Dataset:** MovieLens 100K, including `ratings.csv` , `movies.csv` , and `tags.csv` .

Business Understanding

Modern streaming platforms must recommend content users will enjoy to drive engagement. This system is built to simulate such a platform by learning from historical ratings and genres.

Key Business Questions

- What user behaviors or rating patterns can be leveraged to predict future movie preferences?
- Which recommendation technique offers better predictive performance: memory-based collaborative filtering or model-based filtering
- Can the model generate high-quality, personalized Top 5 recommendations for individual users?

Data Understanding & Cleaning

The MovieLens 100K dataset includes:

- ratings.csv, which contains four columns: userId, movieId, rating, and timestamp, representing a user's interaction with a movie.
- movies.csv, which includes metadata such as the movie title and associated genres.
- tags.csv, which logs user-submitted tags on movies, potentially useful for content-based or hybrid filtering.
- links.csv, which provides external identifiers such as IMDb and TMDb IDs for integration with third-party data.

Data Preparation

- Cleaned datasets which include ratings.csv and movies.csv to produce a structured dataset.
- Filtered to users with ≥ 20 ratings and movies with ≥ 50 ratings, yielding 41,360 ratings.
- Transformed data using the Surprise Reader object.
- Used an 80/20 train-test split to evaluate performance.

Exploratory Analysis

- Found users favor highly-rated movies in Drama, Comedy, and Thriller genres.
- Active users and popular movies provide strong signals for recommendations.

Modeling

Implemented two collaborative filtering models using the Surprise library:

Model	RMSE	MAE
SVD	0.8368	0.6395
KNNBasic	0.8844	0.6753

➡ SVD outperformed KNNBasic on both metrics.

Top Recommendations (User 555)

1. Alien (1979) — ★ 4.55
2. Raiders of the Lost Ark (1981)
3. Clockwork Orange, A (1971)
4. Star Wars: Episode V - The Empire Strikes Back (1980)
5. Terminator 2: Judgment Day (1991)

Hybrid Filtering

- Combined SVD with genre-based cosine similarity.
- Addressed cold-start issues for new users.
- Blended scores with $\alpha=0.7$ (70% CF, 30% content-based).

Insights

- Drama, Comedy, and Thriller are the most frequently rated genres.
- Hybrid models combining collaborative and content-based filtering improve cold-start performance.
- Regular model retraining is important to capture evolving user behavior.

Conclusion

The system combines collaborative and hybrid filtering to deliver personalized movie recommendations, addressing user engagement challenges effectively.

Recommendations

- Deploy SVD-based recommender as the core engine.
- Introduce hybrid filtering for improved cold-start handling.

- Periodically retrain the model on new data.




Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Contributors 3

-  jamesggatonye
-  dantonki
-  Rsumaili

Languages

- Jupyter Notebook 100.0%