

ge-saka / MovieLense

Q

Type / to search

>_

+

▼

<>

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

MovieLense

Public

1 Branch

0 Tags

Q

Go to file

t

Go to file

+

Add file

About

Code

...

<div></div> ge-saka	Update README.md	effc2ef · 3 minutes ago	🕒 26 Commits
<div>content</div>	[+] content csv files	3 hours ago	
<div>.gitignore</div>	[+] .ignore	3 hours ago	
<div>Deployed</div>	Deployed App	1 hour ago	
<div>README.md</div>	Update README.md	3 minutes ago	
<div>RECOMMENDER_S...</div>	recommender	3 hours ago	
<div>app.py</div>	Update app.py	1 hour ago	
<div>movie.jpg</div>	Add files via upload	2 hours ago	
<div>plot.png</div>	[+] memory usage	3 hours ago	
<div>pulp.jpg</div>	Add files via upload	2 hours ago	
<div>requirements.txt</div>	Update requirements.txt	2 hours ago	

DeployedApp

[movieLense.streamlit.app/](#)

Readme

Activity

☆

 0 stars

👁

 1 watching

1 fork

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

 README

RECOMMENDER SYSTEM FOR MOVIELENS DATASET

Summary

This dataset (ml-latest-small) describes 5-star rating and free-text tagging activity from [MovieLens](#), a movie recommendation service. It contains 100836 ratings and 3683 tag applications across 9742 movies. These data were created by 610 users between March 29, 1996 and September 24, 2018. This dataset was generated on September 26, 2018.

Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided.

The data are contained in the files `links.csv`, `movies.csv`, `ratings.csv` and `tags.csv`. More details about the contents and use of all these files follows.

This is a *development* dataset. As such, it may change over time and is not an appropriate dataset for shared research results. See available *benchmark* datasets if that is your intent.

This and other GroupLens data sets are publicly available for download at <http://grouplens.org/datasets/>.

Contributors 2





ge-saka



Pheon-Dev Pheon Dev

Languages

 Jupyter Notebook 99.9%  Python 0.1%

Suggested workflows

Based on your tech stack

**Python package**[Configure](#)

Create and test a Python package on multiple Python versions.

**Django**[Configure](#)

Build and Test a Django Project

**Publish Python Package**[Configure](#)

Publish a Python Package to PyPI on release.

[More workflows](#)[Dismiss suggestions](#)

PHASE 4:GROUP 12 MEMBERS:

Daniel Wahome.

Purity Gitonga.

Brian Kariithi.

Caroline Gesaka.

James Njoroge.

Citation

To acknowledge use of the dataset in publications, please cite the following paper:

F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19. <https://doi.org/10.1145/2827872>

INTRODUCTION

In the era of digital content consumption, personalized recommendation systems play a crucial role in enhancing user experience and engagement. Leveraging the MovieLens dataset from the GroupLens research lab, we aim to build an effective movie recommendation system. This system will utilize collaborative filtering techniques to suggest movies based on user ratings, thereby providing personalized movie recommendations.

BUSINESS UNDERSTANDING

The primary goal of this project is to enhance user satisfaction by recommending movies that align with their preferences. By analyzing user ratings and movie attributes, we aim to create a system that not only suggests popular or highly rated movies but also discovers niche interests that may not be immediately apparent.

PROBLEM STATEMENT

Develop a recommendation system that provides top 5 movie recommendations to users based on their past ratings. The system should address the challenge of sparsity in user ratings and the cold start problem for new users.

OBJECTIVES

Build a Collaborative Filtering Model: Implement a collaborative filtering model to recommend movies based on user ratings and similarities between users.

Address the Cold Start Problem: Explore methods to handle new users with limited or no historical data using techniques such as content-based filtering or hybrid approaches.

Evaluate and Optimize: Evaluate the performance of the recommendation system using appropriate metrics and optimize the model to improve recommendation accuracy and coverage.

METRICS OF SUCCESS

Recommendation Accuracy: Measure the accuracy of movie recommendations using metrics such as Precision@k and Recall@k.

Coverage: Ensure the system can recommend movies across a wide range of genres and user preferences.

User Engagement: Monitor user engagement metrics, such as click-through rates on recommended movies, to assess the system's effectiveness in improving user interaction.

Content and Use of Files

movieId: Unique identifier for each movie in the dataset.

imdbId: IMDb identifier for each movie.

tmdbId: The Movie Database (TMDb) identifier for each movie.

userId_x: Identifier for users who have rated movies.

rating: Rating given by a user to a movie (typically on a scale).

timestamp_x: Timestamp when a user rated a movie.

title: Title of the movie.

genres: Genres associated with each movie.

userId_y: Identifier for users who have tagged movies.

tag: Tags assigned by users to movies.

timestamp_y: Timestamp when a user tagged a movie.

DEPLOYMENT

<https://movielense.streamlit.app/>