

 elsie-n /
Phase4_Project




 Code  Issues  Pull requests  Actions  Projects  Wiki  Security  Insights 





Dataset used is the MovieLens dataset to Build a Recommendation system to predict a future preference list for a certain customer or user, and recommends the top preference for this user.



☆ 0 stars  0 forks  1 watching  Activity

 Public repository

 main ▾

...

 Branches  Tags

 elsie-n ... now 

[View code](#)

README.txt



Summary
=====

This dataset (ml-latest-small) describes 5-star rating and free-text tagging activity from [MovieLens] (<http://movielens.org>), a movie recommendation service. It contains 100836 ratings and 3683 tag applications across 9742 movies.

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.

A recommender system, or a recommendation system, is a subclass of information filtering system that seeks to predict the “rating” or “preference” a user would give to an item. We have two approaches of recommendation systems: Unpersonalized and Personalized. We will however dwell on the Personalized approach in our notebook.

The two types of personalized recommendation systems are content-based recommenders and collaborative filtering systems. We will delve into the application of both in our notebook.

Problem Statement
=====

Our goal is to develop a movie recommendation system that can provide personalized recommendations to users based on their ratings of other movies. By leveraging the MovieLens dataset, we aim to create a model that can accurately identify the top 5 movie recommendations for each user.

Business Objectives

Main objective

Developing a recommendation system.

Specific Objectives

What movie genres are most popular to the users ?

What genres have the most ratings ?

Performance /Evaluation Metrics

Use of RMSE to compare the error of the training and test datasets .

Content and Use of Files

=====

Formatting and Encoding

The dataset files are written as [comma-separated values](http://en.wikipedia.org/wiki/Comma-separated_values) files with a single header row. Columns that contain commas (`,`) are escaped using double-quotes (`"`).

User Ids

MovieLens users were selected at random for inclusion. User ids are consistent between ratings.csv and tags.csv (i.e., the same id refers to the same user across the two files).

Movie Ids

Only movies with at least one rating or tag are included in the dataset. Movie ids are consistent between ratings.csv, tags.csv, movies.csv, and links.csv (i.e., the same id refers to the same movie across these four data files).

Ratings Data File Structure (ratings.csv)

All ratings are contained in the file ratings.csv. Each line of this file after the header row represents one rating of one movie by one user, and has the following format:

userId,movieId,rating,timestamp

The lines within this file are ordered first by userId, then, within user, by movieId.

Ratings are made on a 5-star scale, with half-star increments (0.5 stars - 5.0 stars).

Timestamps represent seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970.

Tags Data File Structure (tags.csv)

All tags are contained in the file tags.csv. The lines within this file are ordered first by userId, then, within user, by movieId.

Tags are user-generated metadata about movies. Each tag is typically a single word or short phrase. The meaning, value, and purpose of a particular tag is determined by each user.

Timestamps represent seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970.

Movies Data File Structure (movies.csv)

Movie information is contained in the file movies.csv.

Movie titles are entered manually or imported from <https://www.themoviedb.org/>, and include the year of release in parentheses. Errors and inconsistencies may exist in these titles.

Genres are a pipe-separated list, and are selected from the following:

- * Action
- * Adventure
- * Animation
- * Children's
- * Comedy
- * Crime
- * Documentary
- * Drama
- * Fantasy

- * Film-Noir
- * Horror
- * Musical
- * Mystery
- * Romance
- * Sci-Fi
- * Thriller
- * War
- * Western
- * (no genres listed)

Data Cleaning

The links and tags dataset seemed to have information contained in the ratings and movies dataset ,the tags column would not be used in our project ,with that a decision to merge the movies and ratings dataset was made to come up with one dataset 'data'.

```
# merging the movies and ratings dataset to have one dataset to work with.
data = pd.merge(movies,ratings,on ='movieId')
```

During the cleaning proces the 'timestamp' column was dropped from the dataset 'data'.

Types of Recommendation systems

There are two types of recommendation systems ;

1. Content-Based

The Content-Based Recommender relies on the similarity of the items being recommended. The basic idea is that if you like an item, then you will also like a "similar" item. It generally works well when it's easy to determine the context/properties of each item.

A content based recommender works with data that the user provides, either explicitly movie ratings for the MovieLens dataset. Based on that data, a user profile is generated, which is then used to make suggestions to the user. As the user provides more inputs or takes actions on the recommendations, the engine becomes more and more accurate.

2. Collaborative Filtering

The Collaborative Filtering Recommender is entirely based on the past behaviour and not on the context.It is based on the similarity in preferences, tastes and choices of two users.

It analyses how similar the tastes of one user is to another and makes recommendations on the basis of that.

For instance, if user A likes movies 1, 2, 3 and user B likes movies 2,3,4, then they have similar interests and A should like movie 4 and B should like movie 1.

These recommendations can be acquired with two broad categories:

Memory-Based Collaborative Filtering (Neighbourhood based).

Model-Based Collaborative filtering.

Assume there are some users who have bought certain items, we can use a matrix with size num_users*num_items to denote the past behaviour of users.Each cell in the matrix represents the associated opinion that a user holds, such a matrix is called a Utility Matrix.

Conclusion:

By leveraging both content-based and collaborative filtering techniques, we were able to develop a model that provided personalised movie recommendations to users based on their ratings on other movies and genres of preference.

Using the collaborative filtering approach specifically user-based, we identified similar users based on their movie ratings and genres which enabled us to generate recommendations based on the preference of users with similar tastes.

We utilized the surprise library which provided a framework for loading and preprocessing the data, splitting it into training and testing sets and implementing the algorithm. Our model was trained using evaluation metrics: RMSE

to access its performance by measuring the accuracy, we ensured that our recommendations were reliable to our users.

By offering the top 5 movie recommendations to the users, we were able to enhance the movie viewage and experience and allow them to discover new films that intrigued them.

Recommendations:

Use of a hybrid recommendation systems that combines content-based and collaborative filtering, hence more accurate recommendations.

Provide a diverse selection of highly popular films that users may enjoy based on the ratings of other movies.

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

● Jupyter Notebook 100.0%