LLD Movie Recommendation System

Introduction

Recommender Systems- We live in an era where every product has thousands of options to choose from. We can take the example of online shopping, gaming, social media and list goes on. Recommender systems help to personalize a platform and help the user find something they like.

Movie recommendation system- A movie recommender system, is an approach to filtering or predicting the users' movie preferences based on their past choices and behaviour. It's an advanced filtration mechanism that predicts the possible movie choices of the concerned user and their preferences towards a domain-specific item, aka movie.

Problem Statement

Given a user id, recommend 5 movies for that user. You should recommend a movie which the user has the highest probability of liking. You must not recommend a movie the user has already seen. Current data set is for 100K ratings. Final data set would be for 1M ratings. So make sure your solution can scale

Overview

Purpose of this document is to cover Low Level Design for Movie recommender system.

Recommender System Workflow

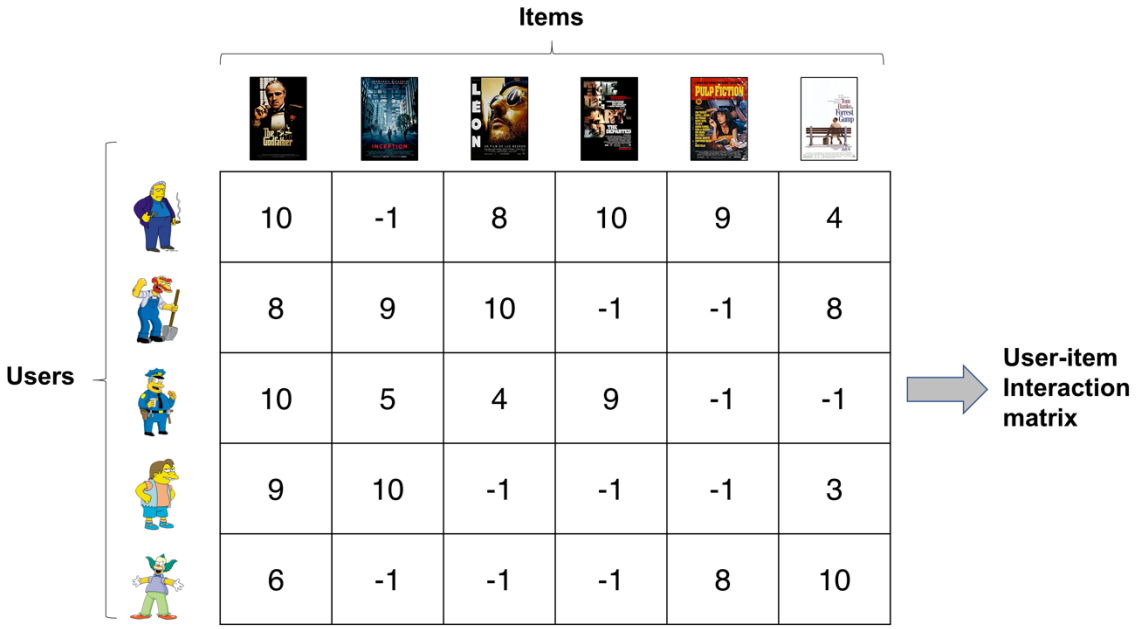
Responsibilities

1. **Data Filtering System**- This system will process raw data and will convert into meaningful information.
2. **Movie System**- It will be responsible for managing movies like storing details of movie, finding relation between different movies on basis of genre.
3. **User Profile System**- It will store data related to user like user attributes and his/her ratings.
4. **Rating System**- This system will be responsible for processing user data, finding most likes movie/ genre.
5. **Genre Management**- It will store available genre types and related movies for all the genre.
6. **Recommendation Algorithm**- This algorithm will be designed with the help of historical data present inside movie, rating system. It will be used to find related movies.

Algorithm Used

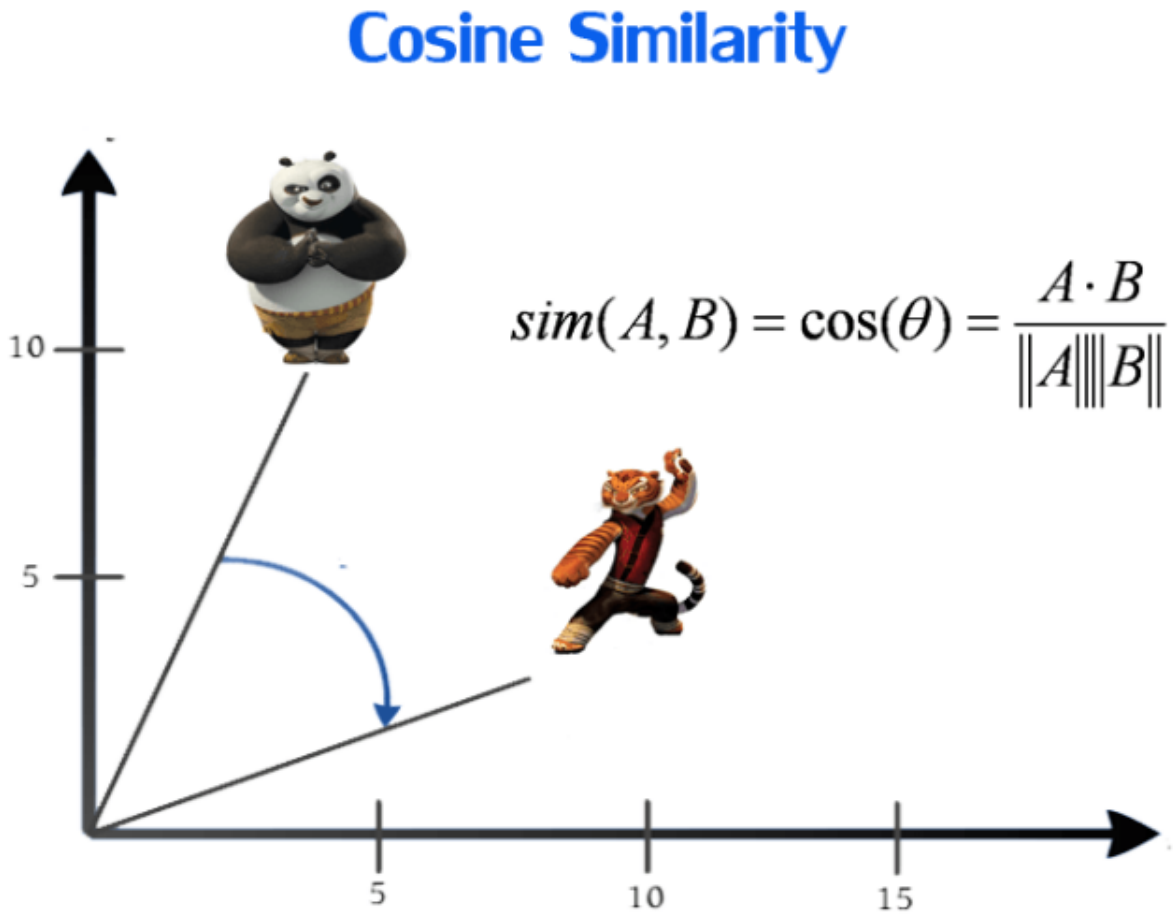
Collaborative Filtering

Collaborative Filtering uses an user-item interaction matrix shown below to extract similarity scores among items or among users. The underlying assumption is:
User based collaborative filtering: users that tend to enjoy many common items would most likely agree on new products.



Cosine Similarity Score

There are a few scores to measure user to user or item to item similarity. Pearson similarity and cosine similarity are the two most commonly used metrics for collaborative filtering algorithm.
Cosine similarity score is the chosen metric for this problem statement. It is the simplest algorithm to find the similarity of two vectors as shown in the image below.



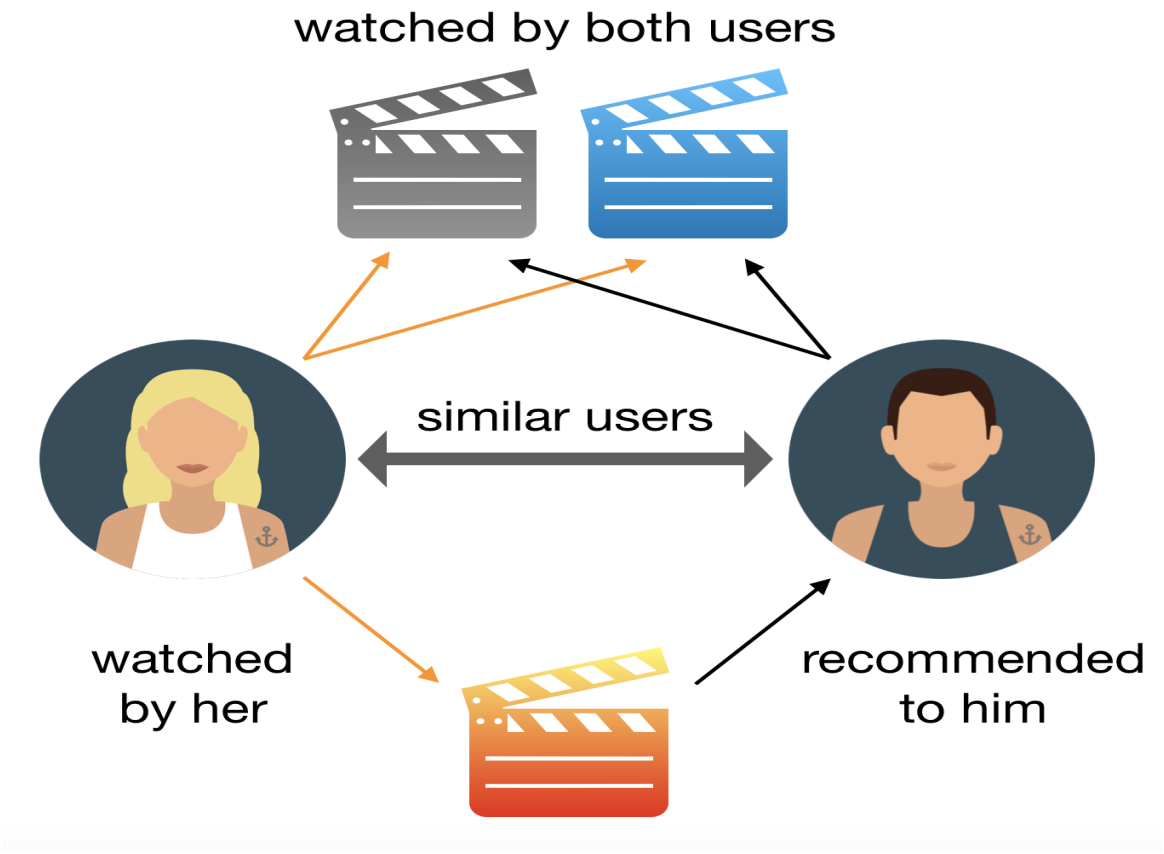
The user score vector (row) or the item score vector (column) can be extracted from the user-item interaction matrix shown above. The cosine similarity score of -1 implies that the users or the items are polar opposites while a score of 1 shows that the users or the items are twins. A score of 0 indicates that the users or the items are completely unrelated.

User Based Collaborative Filtering

This filtering algorithm recommends future items based upon assessing user to user similarity score. The formula for cosine similarity score between user a and b are shown below:

$$Sim(a,b) = \frac{\sum_p (r_{ap} - \overline{r_a})(r_{bp} - \overline{r_b})}{\sqrt{\sum_p (r_{ap} - \overline{r_a})^2} \sqrt{\sum_p (r_{bp} - \overline{r_b})^2}}$$

p in the formula represents all of the commonly rated products by both user a and b. The average rating for both users are subtracted from each rating score to remove any underlying user bias.



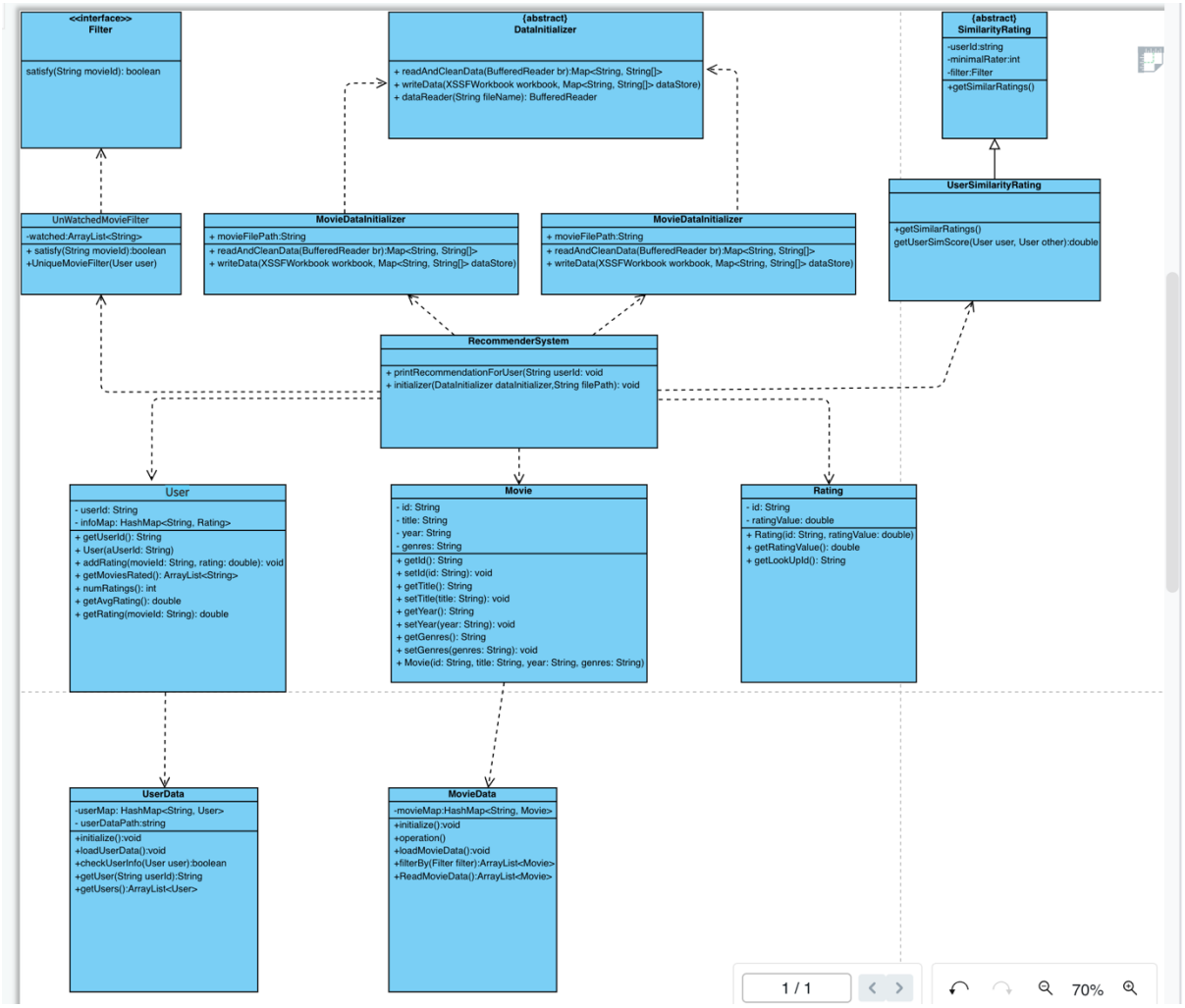
The predicted rating score is calculated using this formula:

$$r_{ap} = \overline{r_a} + \frac{\sum_{j \in users} Sim(a,j) * (r_{jp} - \overline{r_j})}{\sum_{j \in users} |Sim(a,j)|}$$

Design Solution

Layer Name	Class	Notes
Abstraction Layer	Filter, SimilarityRating	
Logic Layer	UserBasedRating	
Persistence Layer	MovieDatabase, UserDatabase	

Class Diagram



Explanation

Movie- This is a model class which have different attribute of a Movie like id, title, year, genre.

Rating- This is a model class which has different attribute of rating like id and ratingvalue. Id can be either userId or movieid as per use case.

User- This class has userId and moviemap attribute and it has a method getAvgRating which is used to calculate average rating of all the movies for a particular user.

RecommenderSystem – It is entry point of the system, it call other classes to start workflows like filtering raw data and initialization of database, and rating calculation.

SimilaratiyRating- It is an abstract class with abstract method getSimilarRatings. It has attribute like userId,minimumRater,Filter.This can be further extend to create Usersimilarrating or Moviesimilarrating.

UserSimilarityRating- It extends SimilaratiyRating to implement userrating similarities so that we can recommend high score movies to user.

Filter- It is an interface which is used to filter movie data on the basis of different criteria.

UnWatchedMovieFilter- It implement Filter interface and provide capability to filter movies data of unwatched movies.

DataInitializer- It is an abstract class with one implemented method dataReader and two abstract method readAndCleanData and writeData.

MovieDataInitializer and **UserDataInitilizer** extends DataInitializer to initialize required data. These classes filter out raw data and convert it to meaningful data.

MovieData and **UserData** store required filtered data of movie and users respectively.

Implementation Details

This project can be implemented as a simple maven project. It will require few maven dependency(commons-csv) to filter raw data and convert it to csv files.

Language- Java
Build Tool- Maven
Database- In memory database (csv file will be used to process data).

Appendix

Open Questions
How are we going to validate the results of recommender system?
How to measure accuracy of the system?

Reference

https://towardsdatascience.com/how-to-build-a-movie-recommendation-system-67e321339109
https://www.geeksforgeeks.org/user-based-collaborative-filtering/

Meeting Notes

Date	Attendees	Notes
15 March 2023	Himanshu Attri, Shivendra Kumar	