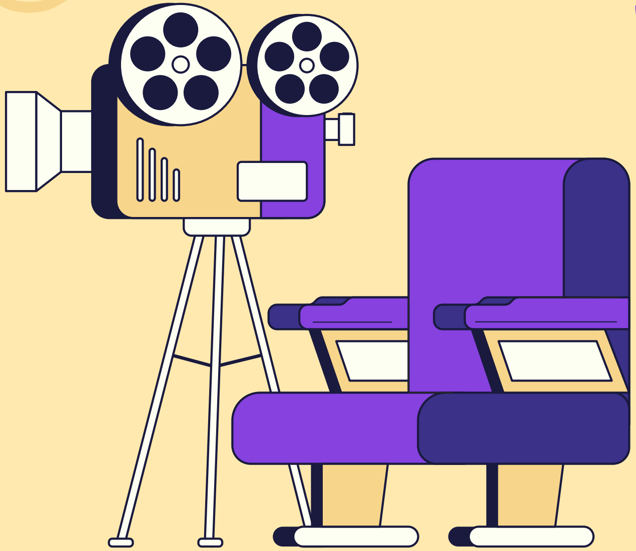


Movie Recommendation System

Eva Burns
05/25/2023

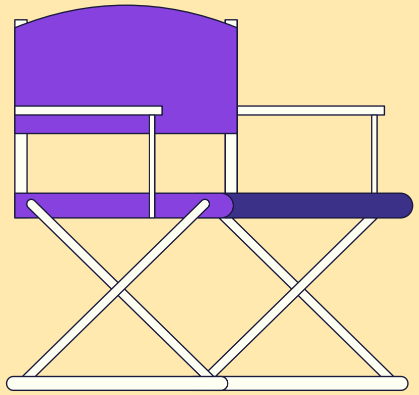
Table of contents

Problem Statement

Everyone has encountered this problem before: you sit down to watch a movie, have no idea what to watch or what you think you would like.

I will create a **recommendation system** using 26 million ratings from 270,000 users.

This will be done by **predicting** what the user will rate each movie and recommending the highest predicted rated movies.



Data Description

The dataset was found at The Movies Dataset on Kaggle

Movies Metadata

Contains information on 45,000 movies found on TMDB.

Relevant features include id, title, genres, popularity, vote_average, and vote_count

Ratings

26 million ratings from 270,000 users for 45,000 movies

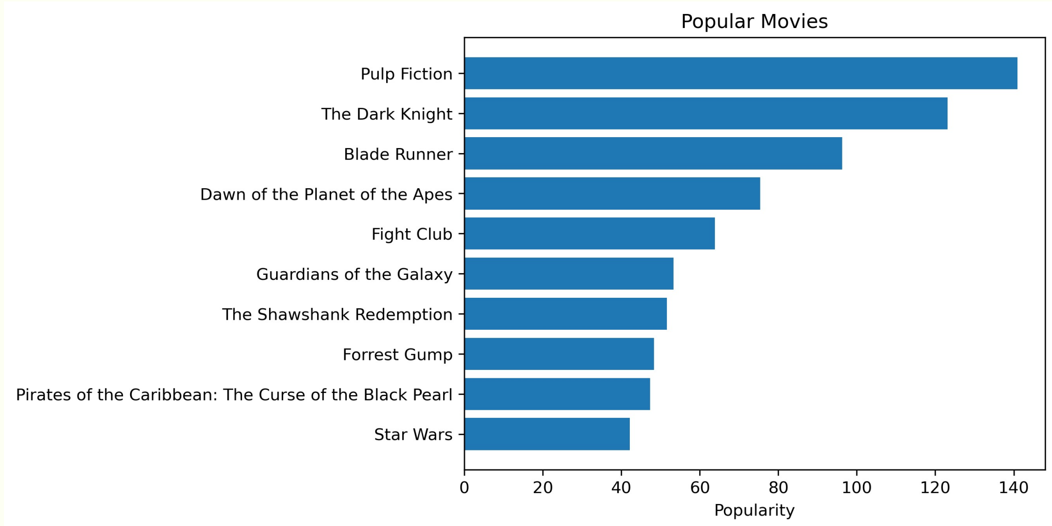
Ratings are from 0.5 to 5, with 5 being the best

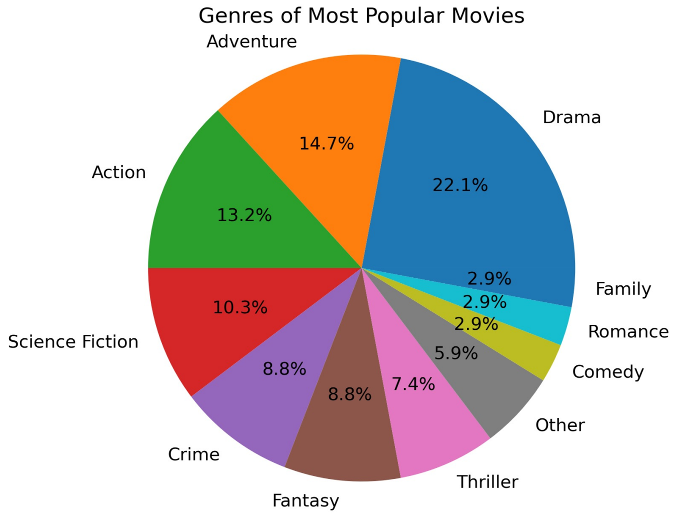
Assumptions/Hypotheses about data and model

- Because ratings are user inputted, there are biases involved in the data
- There are certain groups of people who may leave ratings on movies:
  - Critics
  - People who feel very strongly about the movie (both positively and negatively)
- For the purposes of this project, I will assume that the ratings are representative of the general population

Exploratory Data Analysis – Film Popularity

Most popular movies were found using IMDB's popularity metric



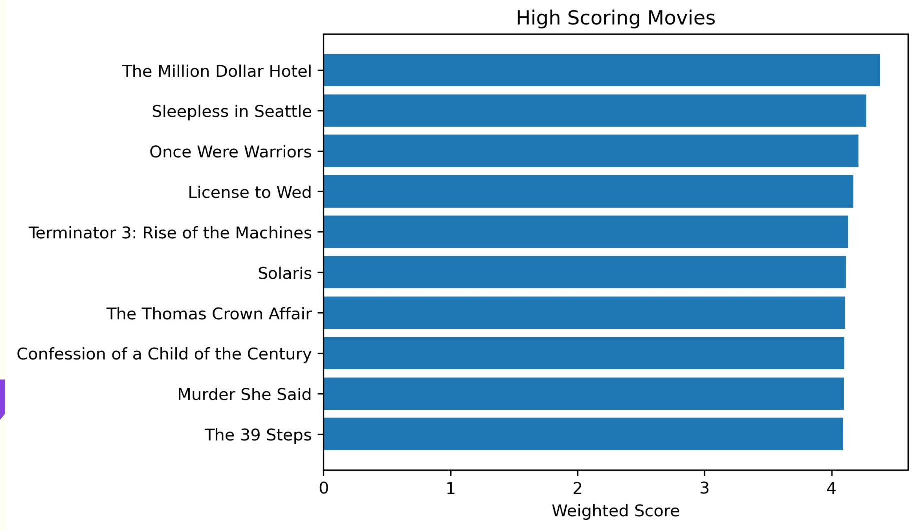


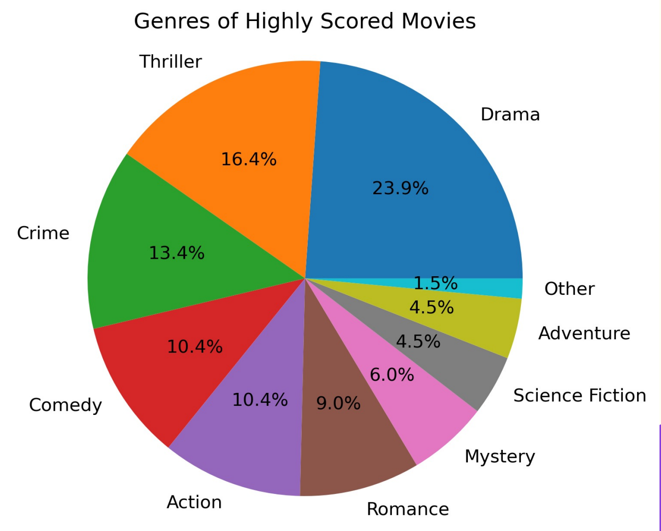
Exploratory Data Analysis – Ratings

Weighted rating was calculated using a formula from IMDB:

$$\textbf{Weighted Rating (WR)} = \left(\frac{v}{v+m} \cdot R\right) + \left(\frac{m}{v+m} \cdot C\right)$$

v - number of ratings for movie, **m** - minimum ratings required to be listed, **R** - average rating of movie, **C** - mean rating across the whole report





Feature Engineering & Transformations

- Convert the ratings data into a readable format for the model I will use which I will explain on the next slide.
- The functions needed for the transformation comes from the Surprise package where the data is converted to a list of tuples
- For example, here are the first ten rows of the ratings dataset converted:

```
[(1, 110, 1.0, None),
(1, 147, 4.5, None),
(1, 858, 5.0, None),
(1, 1221, 5.0, None),
(1, 1246, 5.0, None),
(1, 1968, 4.0, None),
(1, 2762, 4.5, None),
(1, 2918, 5.0, None),
(1, 2959, 4.0, None),
(1, 4226, 4.0, None)]
```

- It is ordered by (userId, movieId, rating)

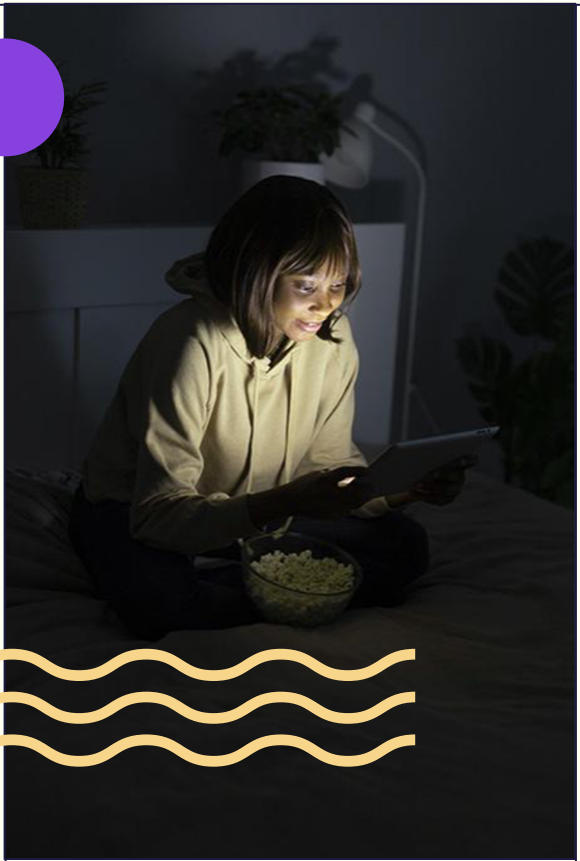
Proposed Approach

- I first thought about using Collaborative Filtering, but it has some limitations:
  - Scalability - computational demands increase as the number of users and movies grows
  - Sparsity - similarity between two seemingly dissimilar movies might be remarkably high due to a single user who ranked them both similarly
- I decided on the **Singular Value Decomposition** (SVD) model found in the Surprise package in python
  - Reduce the dimensionality of the utility matrix by extracting its **latent factors (ratings)**
  - Map each user and movie onto a latent space with a dimension of ‘r’
- Facilitates a more meaningful understanding of the relationship between users and movies, making them directly comparable

Proposed Solution

- Build SVD model, employing a cross validation of 5 folds, using RMSE and MAE as accuracy metrics
- The model **predicts** what a specific user will rate a specific movie
- To make **recommendations** for a specific user:
    - Predict a rating for every movie for the user
    - Sort the predicted ratings to get movies with top predicted ratings
    - Recommend top rated movies

* Note movies already rated by user will be excluded from recommendation



Results (Accuracy)

RMSE - 0.7031

MSE - 0.4943

MAE - 0.5338

Results – Example Recommendation

This is an example of User 1’s recommendation based on their past ratings

Movie	Rating
Sleepless in Seattle	5.0
Rocky Balboa	5.0
Caesar Must Die	5.0
The 400 Blows	4.5
Young and Innocent	4.5
Fools Rush In	4.0
License to Wed	4.0
Shriek If You Know What I Did…	4.0
Confession of a Child of the Century	4.0
The Mystery of Chess Boxing	3.5
Three Colors: Red	1.0

$\longrightarrow$

Movie	Pred. Rating
Beverly Hills Cop III	5.000000
A Woman, a Gun and a Noodle...	4.904216
Rome, Open City	4.895459
Apocalypse Now	4.848280
The Dark	4.840669
Zatoichi	4.793591
While You Were Sleeping	4.784361
We're No Angels	4.777725
Hard Target	4.757174
Red River	4.756867

Future Work

- A **hybrid recommender** incorporating the SVD model with some movie **content** information
  - Based on the movies the user gave a high rating for, find movies that other users also liked
  - Then, find movies that have similar plots, cast, or genre to the movies the user already liked and combine into single recommender
- Also, the highest recommended movie for User 1 was a **sequel**. User 1 has not seen the original movie, so it does not make sense to recommend that movie. An adjustment to the recommender should be made to filter out sequels if the user has not seen the movies before it