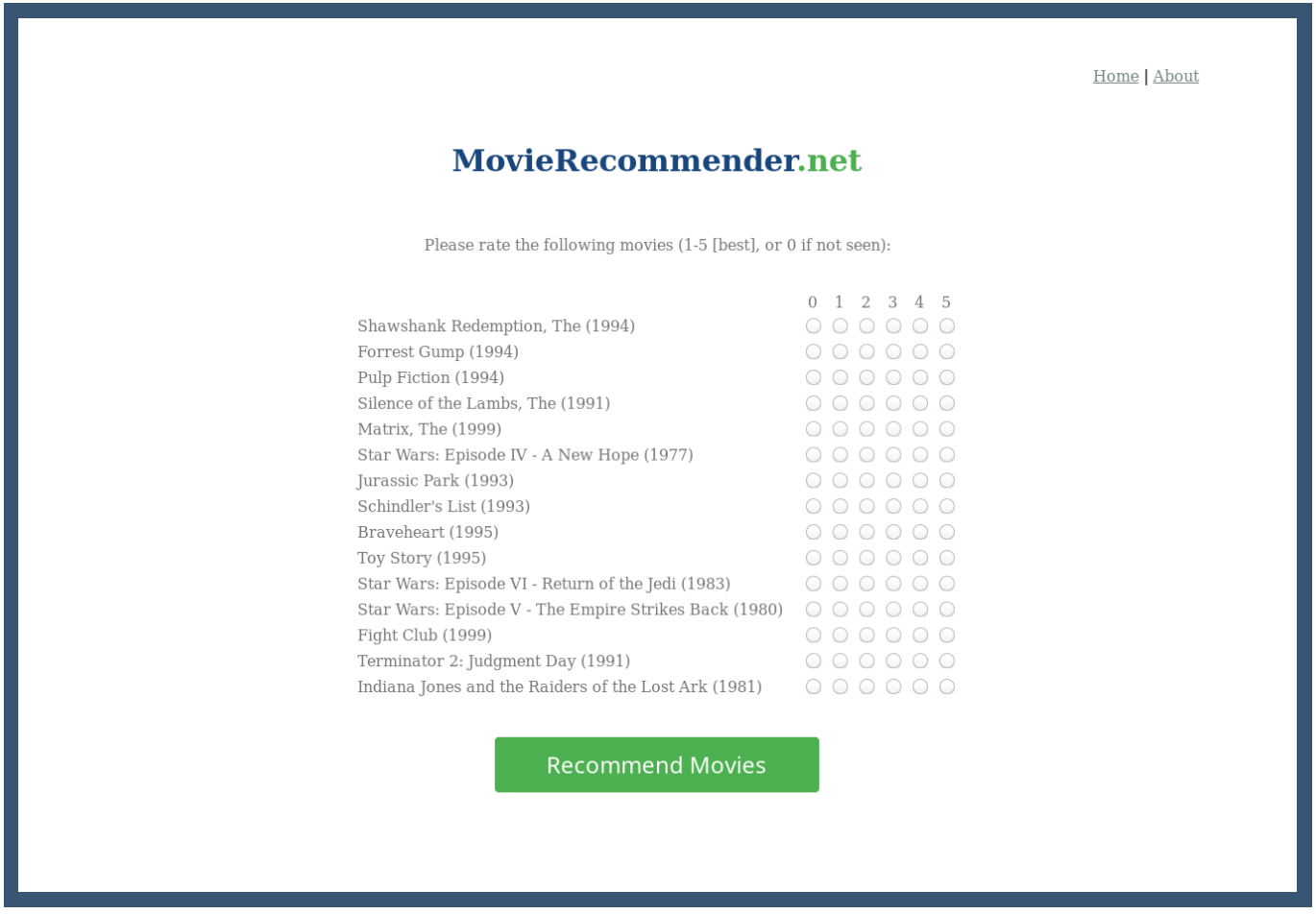
MovieRecommender.net

dn-ds

What is MovieRecommender?

A web app that makes movie recommendations based on ratings supplied by the user.

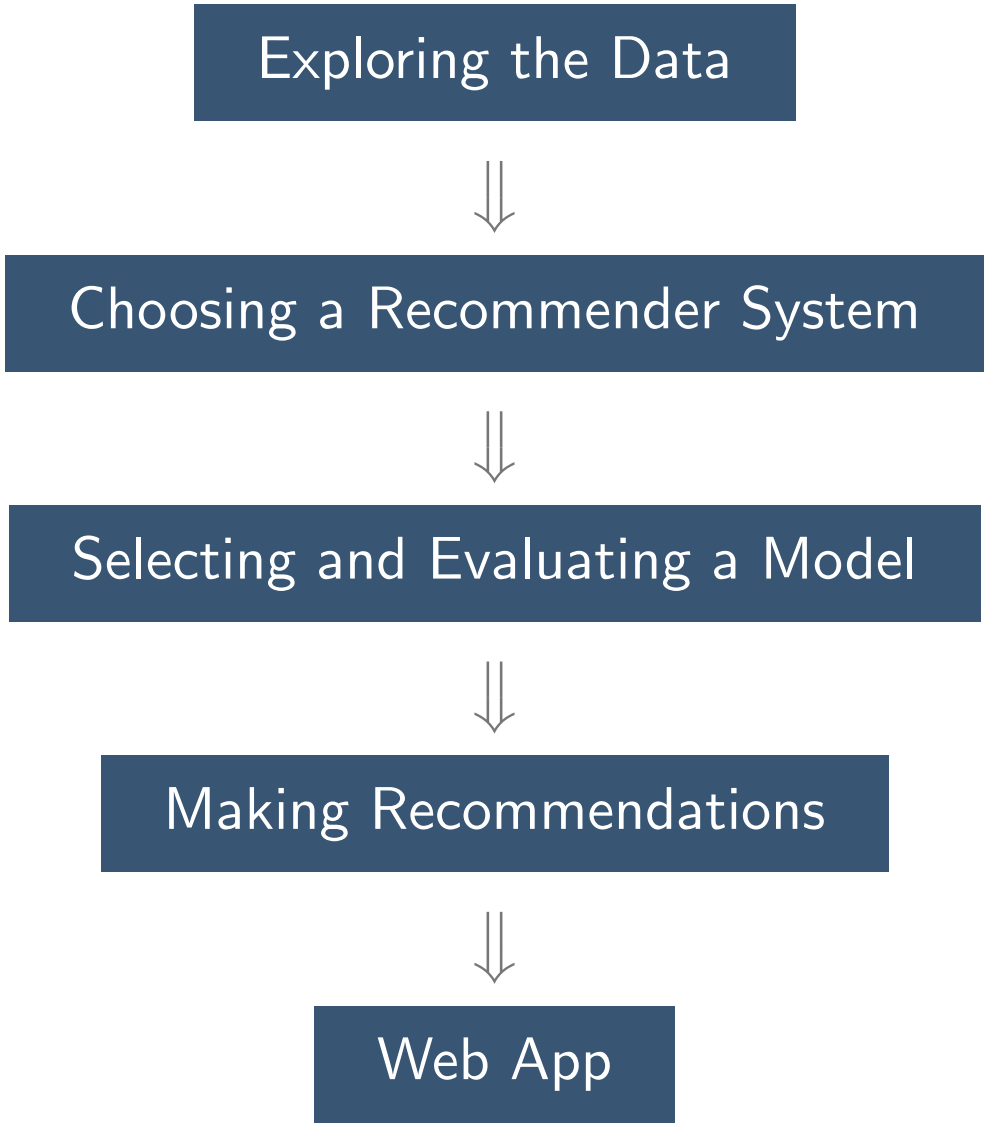


The Dataset

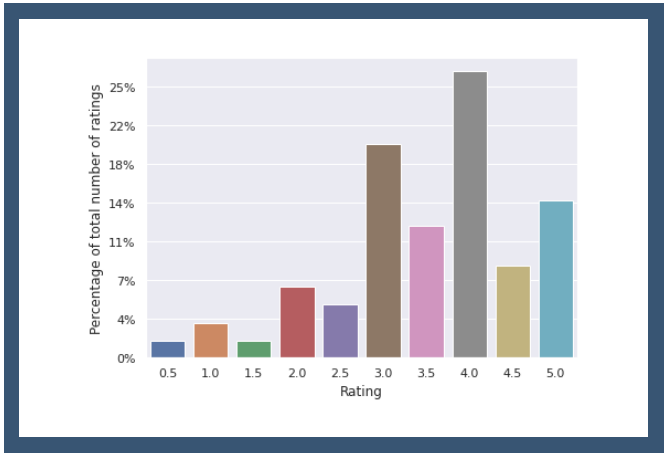
- MovieLens
  `http://grouplens.org/datasets/movielens/latest`
- 27 million ratings
- 280 thousand users
- Ratings created January 09, 1995 – September 26, 2018

Goal: Given user ratings, make movie recommendations.

Project Outline



Exploring the Data



- Ratings range from 0.5 to 5.0, with increments of 0.5.
- Median rating: 3.5; Most common rating: 4.0 (27%).
- # of ratings per user: median $= 30$, min $= 1$, max $\sim$ 24K.
- # of ratings per movies: median $= 7$, min $= 1$, max $\sim$ 98K.
- Training and test sets are prepared.

Choosing a Recommender System

- Used the technique of **collaborative filtering**.
- Main idea: use similarities between users and similarities between items (movies, in our case) simultaneously to provide recommendations.
- Items are recommended (filtering) to a given user based on the interests of similar users (collaborating).
- Used **matrix factorization**, a collaborative filtering algorithm.
- Main idea: decompose the user-movie ratings (sparse) matrix R into a product UV, where U and V are low-rank matrices.

Choosing a Recommender System (continued)

- Optimal U and V are found by minimizing the error arising from the approximation $R \approx UV$.
- Difficult optimization problem, since the objective function is non-convex.
- Instead, solve a convex optimization problem by alternately treating U and V as constants during gradient descent, an approach known as **Alternating Least Squares (ALS)**.

Selecting and Evaluating a Model

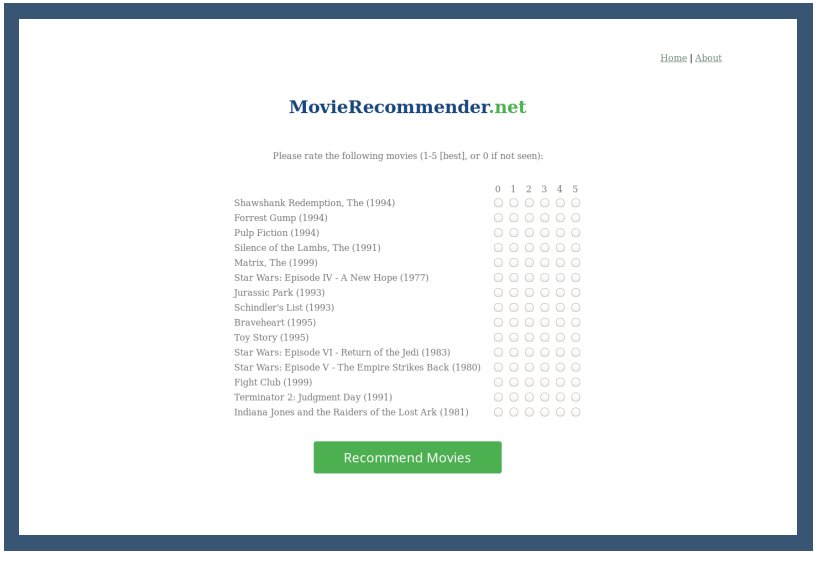
- Used a cross-validator to determine optimal hyperparameters.
- Optimal hyperparameters:
  - rank $= 10$
  - regularization parameter $= 0.1$.
- Root Mean Square Error (RMSE) on the test set: 0.8156.
- Best model improves the baseline model by 15%.

Making Recommendations

- y: ratings supplied by a new user.
- How to make recommendations?
- One approach:
  - Append y to the training set.
  - Retrain the model.
  - Make recommendations.
- Recommendations will not be available immediately.
- Instead, we used the following approach:
  - Find an approximation θ of the user factor vector for the new user.
  - Compute the product θV to obtain predicted ratings for the new user.
  - Make recommendations.

Web App

- When ratings are submitted, the information is preprocessed using jQuery and PHP, and then passed onto the machine learning model.
- The model processes the data and returns recommendations.
- The machine learning model is deployed on an AWS EC2 t2.micro instance using the Flask framework.



Main Tools and Packages Used