

Movie Recommendation

Sheetal Munjewar

DSC680 Applied Data Science

<https://github.com/smunjewar/SheetalM.github.io>

Domain and Background:

Recommender systems, also known as recommendation engines, are tools designed to filter extensive lists of items and suggest those most relevant to a user's preferences. These systems are widely used across various domains, including movies, TV series, and e-commerce platforms. For instance, when searching for similar movies on Google, a recommendation system works behind the scenes to generate a curated list of options.

The dataset used for this project is provided by MovieLens, a trusted and widely utilized platform for movie ratings and tagging. It contains a comprehensive collection of 25 million ratings and over 1 million tag applications across 62,000 movies. The data is well-organized and structured, making it highly suitable for in-depth analysis.

Research Question Quality:

The research questions focus on building a recommendation system to enhance user experience by providing personalized movie suggestions. These questions align with real-world challenges, such as reducing search time and improving user satisfaction on platforms like Netflix and YouTube.

Proposal Grammar and Organization:

The proposal is well-structured and organized, with a logical flow from background information to research questions, methodology, and potential challenges. Grammar is clear and concise, making it easy to follow.

Ethical Considerations – Domain and Background:

Recommender systems are essential for filtering large datasets and presenting relevant suggestions to users. Their application spans movies, e-commerce, and other industries. It's crucial to ensure fairness and avoid biases in recommendations, as these systems significantly influence user choices.

Recommendation Engines Overview:

1. **Content-Based:** Utilizes attributes of items (e.g., movies) to recommend similar ones.
2. **Collaborative Filtering-Based:** Uses user activity (e.g., ratings, reviews) to find similar preferences.
3. **Popularity-Based:** Suggests items based on overall popularity.
4. **Hybrid:** Combines two or more recommendation methods for improved performance.

References:

- Sharma, A. (2020). *Beginner Tutorial: Recommender Systems in Python*. DataCamp.- <https://www.datacamp.com/community/tutorials/recommender-systems-python>
- Yurtekin, Ş. *Movies Recommendation*. Kaggle. <https://www.kaggle.com/sevvalyurtekin/movies-recommendation>
- Wikipedia Contributors. *Recommender System*. Wikipedia. https://en.wikipedia.org/wiki/Recommender_system#:~:text=A%20recommender%20system%2C%20or%20a,would%20give%20to%20an%20item.
- Rocca, B. (2019). *Introduction to Recommender Systems*. Towards Data Science. <https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>
- Isinkaye, F. et al. (2015). *Recommendation Systems: Principles, Methods, and Evaluation*. <https://www.sciencedirect.com/science/article/pii/S1110866515000341>
- Building a Movie Recommendation Engine in Python using Scikit-Learn – Suman Adhikari – Feb 2019 - <https://medium.com/@sumanadhikari/building-a-movie-recommendation-engine-using-scikit-learn-8dbb11c5aa4b>
- Build A Movie Recommendation Engine Using Python - Aug 2019 - <https://medium.com/analytics-vidhya/build-a-movie-recommendation-engine-using-python-scikit-learn-machine-learning-e68ba297e163>
- Movies Recommender System - Rounak Banik - <https://www.kaggle.com/rounakbanik/movie-recommender-systems>
- recommendation through movie_lens_rating - Karan Choudhary - <https://www.kaggle.com/karanchoudhary103/recommendation-through-movie-lens-rating?select=tag.csv>
- An In-Depth Guide to How Recommender Systems Work - Badreesh Shetty – Jul 2019 - <https://builtin.com/data-science/recommender-systems>

Data Understanding:

The dataset includes:

1. **Ratings File:** userId, movieId, rating, timestamp.
2. **Tags File:** userId, movieId, tag, timestamp.
3. **Movies File:** movieId, title, genres (pipe-separated).

Source: [MovieLens Dataset](#)

Research Questions:

With the growing number of movies on platforms like Netflix and Disney+, a recommendation system is essential for saving users' time and enhancing their experience. How can a hybrid recommendation engine improve suggestions compared to single-method engines?

Methodology:

- Start with a **Collaborative Filtering-Based** recommendation system.
- Progress to a **Hybrid System** combining collaborative filtering and content-based techniques for enhanced results.

Potential Issues:

1. Limited experience with recommendation systems, requiring a learning phase.
2. Large dataset size might present hardware challenges.
3. Time constraints may limit the completion of a hybrid system.

Concluding Remarks:

Recommender systems are vital for enhancing user engagement on platforms like Netflix and YouTube. By leveraging machine learning, they save time and improve user satisfaction. Developing a robust system can keep users engaged and loyal, ensuring a seamless and personalized experience.