Problem:

In this project, we delve deep into movie recommendations by analysing diverse datasets from ImBD, Movie Lens, Kaggle, Rotten Tomatoes, and Netflix Our primary objective is to amplify the efficiency of movie recommendation systems to enhance the media viewing experience. We aim to transform the movie viewing data into a customer-centric dataset. We intend to develop a recommendation system to suggest the best movie for a given profile to customers within each segment who have yet to see these films, ultimately enhancing efficacy and fostering increased customer satisfaction.

Objectives:

- Data Cleaning & Transformation: Clean the dataset by handling missing values, duplicates, and outliers, preparing it for effective modelling.
- Feature Engineering: Develop new features based on the viewing data to create a customer-centric dataset.
- Recommendation System: Implement a system to recommend the most appropriate movies to users within the same viewing profile who have yet to view those movies to boost customer experience.

Datasets overview:

- Netflix - publicly available dataset released by Netflix for an optimisation challenge
- Movie Lens - publicly available data set
- Kaggle Challenge - publicly available dataset uploaded to Kaggle
- Rotten Tomatoes - publicly available data set
- Imbd - titles and ratings publicly available dataset
- APIs : ombd and tmbd, publicly available APIs

Columns overview:

- User profile (from user input):
    - userID
    - Name
    - Username
    - Email
    - Password
- User preferences (from user input):
    - Genres
    - age rating
    - user ratings
    - release year range
- CSV data from datasets stored in SQLAlchemy database:
    - movieID
    - title,
    - yrmade,
    - isAdult,
    - runtime,

    - Genres,
    - ratingavgscore,
    - actors,
    - company,
    - agerating,
    - tags,
    - votes,
- Data pulled from the API:
    - Title
    - genre(s)
    - rating
    - run time
    - age rating

Conclusion

This project successfully created a recommendation system for movie viewing. Going forward it would be great to use web scraping to further enhance the dataset and to be able to spend more time on data engineering to better extract, transform and load the data sets.

Appendix

Formulas used:

Cosine similarity

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}^\intercal}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} = \frac{\sum_{i=1}^{n} \mathbf{x}_i \cdot \mathbf{y}^\intercal{}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{x}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{y}_i)^2}}$$

TF IDF algorithm

$$TF - IDFscore(w_{ij}) = TF_{ij} * IDF_i$$

Algorithm visualised: