COMP9727 Project Design

The Design Scheme of

The Movie Recommendation System

FULL NAME: HAOCHENG YANG

ZID: Z5532140

E-mail: z5532140@ad.unsw.edu.au

Submission Date: 03/07/2025

1 Project scope

1.1 Project background and Motivation

With the emergence of digital streaming today, video websites on the internet hold massive movie libraries and accumulate massive-scale user behavior data. How to provide dynamic personalized movie recommendations among tens of thousands of movies in consideration of users' history preferences, real-time watching interests, and genre rules is currently a fundamental research and application trend in recommender systems. Personalized recommendation not only enhances user engagement and site stickiness, but also possesses significant business advantage in maximizing the conversion rate of high-margin content along with long-tail movie assets' visibility.

The objective of this project is to design and implement a real-user-behavior-based movie recommender system for the general recommendation context of "daily movie recommendations." The system will simulate the actual interaction process of the user while browsing and watching movies, build individualized interest profiles, and use multi-dimensional information such as history of viewing, movie content features, and user preferences to generate personalized and accurate recommendations. The recommendation result will be presented on a web interface. Users can provide feedback by clicking to view, skip, rate, or save to watchlist, thereby enabling the system to further improve recommendation quality.

1.2 Domains and Target Users of the recommendation System

The project belongs to the domain of "online movie platforms." Target audience is casual platform users watching movies using online streaming services, specifically active user segments with consistent engagement behavior and definite viewing habits. They typically want to discover new movies, re-watch previously viewed movies, or look for movies that suit their mood, genre interest, or viewing situation. Therefore, the recommendation system must maximize both diversity and accuracy, providing content that is exploratory and relevant.

1.3 Recommendation Scenarios and Recommendation Forms

The system supports the following three primary recommendation scenarios:

- Daily Movie Recommendation: It recommends 10 individualized movie titles to customers on a daily basis based on past viewing behavior and present interest.
- Similar Movie Recommendation: It recommends other movies featuring the same genres, themes, or styles of narrative after users view a particular film.
- Thematic Playlist Recommendation: Suggests movies or watchlists of interest or emotional state to users.

Recommendations are also displayed in card-based format on the web interface. Each card displays movie title, poster image, director and cast name, synopsis preview, runtime, and interaction buttons. Users can click watch, save later, skip, or indicate disinterest.

1.4 User Interaction Methods and interface sketches

Users interact with the recommendation system through the following activities:

- Click "Watch Now" or "Skip" to record behavioral preference;
- Click "❤️" or "Add to Watchlist" to record interest or willingness to view, which is used for preference modeling;
- Flag content as "Not Interested" to provide negative feedback;
- Visit recommendation homepage once a day will refresh a new list of 10 personalized movie recommendations.

The layout of the interface (as seen from Figure 1) is a simple, intuitive design intended for good feedback collection.

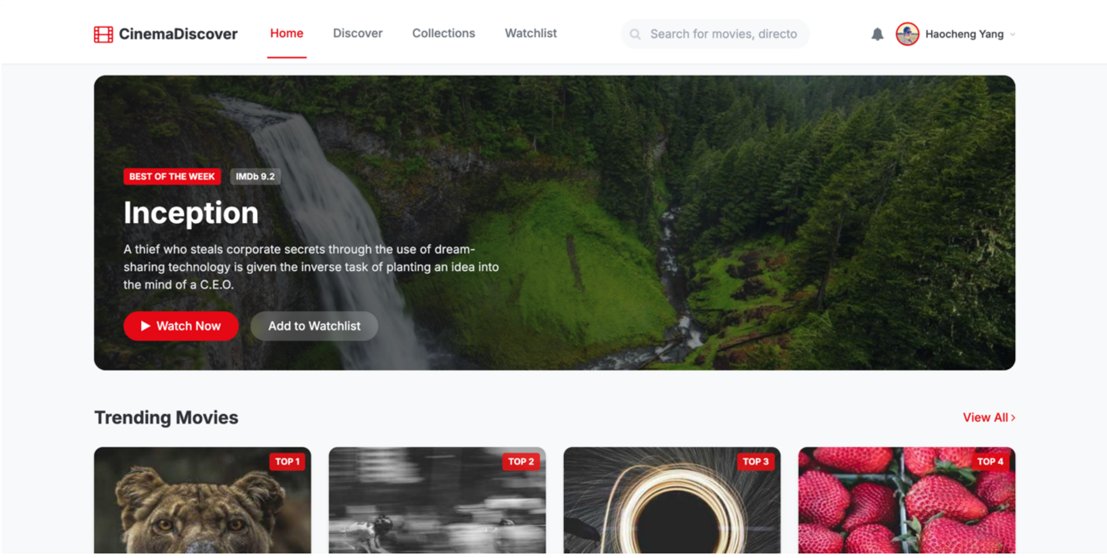

Figure 1 Suggested page display image(self-designed)

1.5 Recommended update mechanism and cold start strategy

To counteract user interest drift and increasingly large movie collections, the recommendation process utilizes a hybrid update mechanism of offline batch training and online incremental updates, as well as integrated cold-start strategy for improved responsiveness and stability.

- Model Update Mechanism
  - Offline Batch Training (daily/weekly):
    - Periodic retraining of collaborative filtering and ranking models over the complete user interaction logs ensure high-quality global recommendations.
  - Online Incremental Update:

        - Real-time updating of user profiles using the most recent view activities. Session-level interest modeling by light-weight content-based modules enhances short-term recommendation precision.

- Cold Start Processing Strategy
    - Cold Start for New Users:
        - Allow onboarding questionnaires to enter liked genres, favorite directors/actors, and movie themes to seed user profiles.
        - Guide users through a short onboarding viewing interaction to accelerate profile convergence.
        - Use demographic-based clustering for preference seeding from population-level modeling.
    - Cold Start for New Movies
        - Balance content vectors from metadata such as genre labels, cast, keywords of synopsis, and release date.
        - Utilize content-based similarity measures to project user profiles on to new films.
        - Embed exploratory recommendation spaces in which new films are introduced in a subtle manner in order to elicit feedback for application in collaborative models.

1.6 Rapid Business Model Analysis

Following are key business values and commercial advantages that might not need dramatic enhancement with a movie recommendation system:

- Enhanced User Engagement: Personalized recommendations encourage users to watch more frequently, explore the platform's catalog, and return frequently.
- Enhanced Subscription and Advertising Revenue: Real-time and accurate recommendations maximize interaction with paid content and ad exposure efficiency.
- Exposure to Niche and Licensed Content: The system makes lesser-known films or newly licensed content visible to suitable users, enabling content discovery beyond blockbuster content and boosting long-tail content use.

2 Dataset Design

2.1 Data Sources and Explanations

The project uses real user behavior data pertaining to the movie recommendation task, which has been extracted from public datasets. The following dataset is selected as primary sources:

- IMDb + TMDb Metadata

This combined source holds dense movie metadata including genres, cast, directors, overviews, keywords, poster images, and release dates. Together with the dataset via movie IDs or titles, it supports effective utilization of content-based recommendation models and enables cold-start scenarios.

2.2 Data Content and Field Interpretation

The datasets hold key fields that indicate user preference, movie content features, and temporal viewing patterns:

2.2.1 User behavior information:

user_id	Unique user ID
movie_id	Unique movie ID
rating	User rating
timestamp	Time of rating or interaction

Table 2.2.1

2.2.2 Movie content information:

title	Movie title
genres	List of genre tags
overview	Brief plot description
director	Key actors and directors
release_year	Year of release

Table 2.2.2

2.2.3 User attribute information:

age	Age
gender	Gender
location	Position

Table 2.2.3

2.3 Data Quality and Preprocessing

For ensuring the accuracy and usability of the recommendatory models, this project preprocessor systematically raw data by applying data filtering, normalization, sparsity management, behavioral conversion, and time-division segmentation, as described below:

2.3.1 Sample size and scale analysis

- User scale: The data set contains more than 138,000 users who rate items with varying behavior, supporting personalized preference modeling.
- Item scale: The data set contains more than 27,000 movies, and exploration of mainstream as well as niche material in the recommendation domain is possible.
- Interaction volume: More than 20 million interactions are present, providing enough density for training collaborative filtering and hybrid models.

2.3.2 Sparse control and data cleaning

- User activity filtering: Exclusion of users with less than 10 ratings to limit noise and provide stable training signals.
- Movie popularity filtering: Exclusion of movies with less than 20 overall ratings to alleviate sparsity and long-tail extremes.
- Duplicate removal: Merge movie title duplicates by ID/title mapping.
- Missing field elimination: Exclude records without vital fields like movie_id or rating.
- Outlier cleanup: Remove invalid or out-of-range ratings or timestamps.

2.3.3 Behavioral normalization and discretization

- Rating binning: Discretize ratings for binary preference modeling or conversion to implicit feedback.
- Weighted preference scores: For hybrid models, combine rating scores with recency or frequency of interaction to compute an enhanced preference weight.
- Log transformation: For interaction counts, employ log normalization.

2.3.4 Data partitioning strategy

- User-level split: Assign all users training, validation, and test set interactions to prevent cold-start during test time.
- Time-based split: Use timestamps to generate narratives train/test splits.
- Interaction balance control: Maintain uniform user activity and movie popularity distributions across data splits to prevent evaluation unfairness.

2.3.5 Processing and enrichment of content field

- Text preprocessing: Plot summaries are lowercased, punctuation removed, and cleaned for TF-IDF or embedding model input.
- Genre/tag encoding: Genres and keywords are encoded using multi-hot vectors or embeddings to be applied in similarity computation.
- Release period categorization: The movies are grouped based on release decade to allow for temporal preference analysis or clustering.
- Feature fusion: Movie content vectors are mapped against user preferences of genres to

enable cold-start and hybrid recommendation.

2.4 Analysis of the Limitations of the Dataset

- Restricted behavioral dimensions and lack of explicit negative feedback
  - IMDb has predominantly positive rating signals and lacks negative interactions such as "dislike," "skip," or "not interested."
  - This limits the model from distinguishing between "not seen" and "seen and disliked" cases, especially in Top-N recommendation scenarios.
  - Solution: Add a simulated recommendation environment where users provide active feedback to enhance training data with clear preference signals.
- Incomplete or inconsistent content fields impact content modeling
  - Completeness of metadata varies across movies; very few have blank or default genres, vague summaries, or conflicting tag annotations.
  - This reduces the content-based recommendation and cold-start performance resilience.
  - Solution: Use NLP-based techniques to enhance plot features; fill in missing fields using TMDb API or IMDb scraping.
- No actual real-time user-system interaction loop exists; hard to simulate end-to-end deployment
  - Public data doesn't simulate the real user-system interaction loop; click-through rate, dwell time, or satisfaction measures aren't present.
  - Qualitative aspects of user experience cannot be received directly from logs.
  - Solution: Build a simulation interface with real users performing recommendation tasks and provide post-session questionnaires; compare model variations by A/B testing in a controlled environment.

3 Recommended Method

3.1 Method Selection Overview

The Hybrid Recommendation System framework employed in this project is an integration of three classes of recommendation methods: collaborative filtering, content-based recommendation, and sequence modeling optionally. The approach primarily leverages user behavior logs, movie content features, and temporal preferences to improve performance on key directions such as accuracy, coverage, diversity, and cold-start robustness.

3.2 Content-Based Recommendation

The content-based module utilizes both structured metadata and unstructured text descriptions of movies to build high-dimensional feature representations:

- Movie Vector: Genres, plot summary, and keywords of a movie are cleaned and represented in terms of TF-IDF or semantic embedding models.
- User Profile: For a specific user, the preference vector is built by summing up features of liked or rated movies, recency or rating strength as optional weights.
- Recommendation Strategy: Compute cosine similarity between user profile vector and candidate movie vectors, and return the Top-N most similar movies as the recommendation.

This module is especially useful for cold-start scenarios, especially for new users or movies with not enough collaborative data.

3.3 Collaborative Filtering Recommendation

This module leverages a latent factor collaborative filtering method to uncover implicit preference relationships between users and movies:

- Construct a user-movie interaction matrix from explicit ratings or implicit signals.
- Apply matrix factorization algorithms such as SVD or ALS to complete the sparse matrix and learn user and movie latent representations.
- For each user, compute the inner product between his/her latent vector and all unseen movie vectors, rank them, and recommend the Top-N highest rated movies.

This method proves powerful in catching underlying affinities between similar user profiles and unseen but relevant items, ideally suited for medium-to-heavy users.

3.4 Hybrid Recommendation Mechanism Design

To take the strengths of both collaborative filtering and content-based models, the system applies two complementary hybrid methods:

- Weighted Fusion Model:Linearly combine both models' scores with an adjustable weighting factor α:

$$final\_score = \alpha \cdot score\_CF + (1 - \alpha) \cdot score\_CB$$

In this case, α is adjusted between exploration precision and content adaptability so that it can be utilized for exploration-exploitation trade-off, especially in cold-start cases.

- Two-Stage Ranking Model:
  - Phase 1 (Candidate Set Generation): Use collaborative filtering or content similarity to produce a Top-K candidate set efficiently.
  - Phase 2 (Sorting and Fine Ranking): Extract multi-dimensional features of each candidate, and employ a light-weight learning-to-rank model such as Logistic Regression or XGBoost to re-rank the candidate set.

3.5 Analysis of Method Applicability and Feasibility

- Good Data Matching: The data set provides rich user-movie interaction records suitable for collaborative filtering schemes. IMDb/TMDb metadata supports good content modeling for genres, actors, plots, and keywords. Both sources complement each other well, especially for hybrid layout.
- Clear Task Objectives: The framework is designed to address Top-N movie recommendation with personalized ranking tasks. The architecture facilitates clear metric alignment and relevance-based testing based on user profiles and past viewing habits.
- Scalable and Modular Architecture: All components are loosely coupled, allowing for future additions like:
  - Context-aware recommendation;
  - Online A/B testing for comparing various ranking models;
  - Swappable ranking modules and feature pipelines.

4 Model and System Evaluation

4.1 Model Evaluation Indicators (Based on Historical Data)

The recommendation model will conduct static performance analysis in the offline system per user basis based on user behavior logs by the following metrics:

- Precision@N: It calculates the ratio of top N items in the recommended list that the users favor in actuality, representing the hit ratio of recommendations.
- Recall@N: Calculate the coverage of the top recommended results in the material which the users are assumed to like as an indicator of recall ability of the suggestions.
- F1@N: The harmonic mean of Precision and Recall to achieve a trade-off between precision and coverage.
- MAP: Calculate the average of a number of users' Precision curves to emphasize the overall ranking quality of the suggested list.
- NDCG@N: Adds position weights to increase the ranking scores of highly ranked recommendation systems for extremely relevant items.
- User Coverage: The proportion of users recommended by the system and hit at least one item, which indicates the extensive coverage and fairness of the system's services.
- Novelty and Serendipity: "Novelty" captures the level of novelty of recommendation outcomes and tends to suggest something new to the user. "Serendipity" determines if the recommendation outcomes are "unexpected but surprising" and is used to capture the capability and diversity of exploration of recommendations.

4.2 Recommendation System Evaluation Indicators (Based on User Feedback)

The system testing at the user level is conducted in the interactive simulation environment for the purpose of verifying the quality of recommendation performance while in actual use. The most important dimensions of assessment are:

- Click-through rate: Measures how often users click on "Watch" or "View Details" for recommended movies, expressing initial interest or salience.
- Skip rate: The ratio of movies that users intentionally skip or bypass, indirectly reflecting poor recommendation matching.
- Dwell time on movie pages, used to infer perceived content quality.
- Conversion rate: Quantifies more intense user activity such as "Add to Watchlist," rating submission, or full view.
- Satisfaction survey: Captures user-perceived suggestion quality via 5-point Likert scale.
- User feedback keywords: Requests users to describe browsing or view experience in short open-ended responses for qualitative inspection.

4.3 Multi-Indicator Weighing and Selection

In order to further utilize the practical effect of personalized recommendation, this project will set the following core optimization goals for multi-indicators:

- Precision@N and MAP: As primary indicators used to evaluate recommendation quality and ranking quality, they will be used as the model's first-choice indicator for tuning to maximize the precision of Top-N movie recommendation.
- F1@N: Enables a trade-off between precise targeting and large retrieval coverage, reducing overfitting towards certain interests;
- User Coverage: Makes sure that tailored advice is delivered to many users, not merely heavy users or specific segments.
- Novelty and Serendipity: Promotes a combination of pertinent and unexpected suggestions—assisting users to discover movies they would not otherwise have encountered—thus increasing satisfaction as well as discovery.
- NDCG@N: Optimizes appropriately ranking the most relevant films at the head of the recommendation list, an important consideration in UI-constrained environments.

The final model selection will weigh the performance of the said indicators above and be optimized and blended with actual user comments to ensure the recommendation system a good trade-off among diversity, accuracy and interpretability.

4.4 Real-time Performance and System Performance Considerations

Since "Daily Movie Recommendation" is triggered mostly for non-real-time instances, the system has comparatively low response latency requirements, but nevertheless needs to ensure efficiency of operation and scalability:

- Recommendation calculation mode: Precomputed daily recommendation lists using offline batch processing and front-end display is served via caching.
- Lightweight online update: Real-time updating of user profiles online for usage in the subsequent round of recommendation optimization.
- Model update frequency: Update the master model weekly, and the built-in sorting model can be updated on a daily basis.
- Pre-generation recommendation mechanism: Pre-generate online users' recommendation results during the night and push them to the database for maximum system throughput.

4.5 Simulation User Test design

Since no closed-loop feedback is available in public data, this project will only perform a small-scale simulated user test experiment. The process is as follows:

- Experimental platform: A light web-based or notebook-based interface will be built to accommodate real-time user interaction with movie suggestions.

- Task process:
  - Users log in and optionally complete a brief preference questionnaire;
  - The system generates a daily list of 10 movie suggestions, on which users can interact (click, skip, rate, add to watchlist);
  - After interaction, users complete a brief feedback form with a Likert-scale test and optional short answer section.
- Collect information:
  - Interactive Log
  - User satisfaction rating
  - Subjective feelings description

With the objective behavior and subjective feedback in hand, usability and practicability of the recommendation system are tested to see if it provides reasons for further team optimization.

5 Conclusion

This paper proposes a hybrid recommendation framework specifically for personalized movie recommendation. By combining collaborative filtering and content-based modeling, the framework can effectively address significant challenges such as cold-start scenarios and user interest drift. By the careful selection of high-quality public datasets and adherence to a structured preprocessing and evaluation pipeline, the proposed model is designed to possess excellent accuracy and scalability.

In the subsequent stage, the team will deploy systems and conduct simulated user testing, continuing to enhance performance of recommendations and refine the user experience. This will facilitate the building of a robust, responsive, and user-centric movie recommendation system from real usage behavior.