BEIJING INSTITUTE OF TECHNOLOGY

Movie Recommendation System Based on Multi-Component Scoring

Project Title: Movie Recommendation System Based on Multi-Component Scoring

Course: Programming with Python (Fall 2025)

Date: November 2025

Members:

Name	Student ID
许双博	1120243389
王韵嘉	1120240891
武红玮	1120243412
李尚泽	1120242314

content

1. Introduction & Motivation

The movie industry produces thousands of films each year. As streaming platforms and online libraries continue to grow, finding worthwhile movies has become increasingly challenging. Algorithms that recommend films based on a single metric (e.g., popularity or genre) fail to capture the multi-faceted nature of what makes a film appealing. Our project addresses this challenge by blending several complementary attributes—content relevance, audience reception, business success and timeliness—to produce balanced recommendation scores. The goal is to design a transparent and reproducible pipeline that processes a large public movie dataset and generates ranked lists suitable for different user personas.

2. Problem Definition

Given a collection of 4 802 films with metadata such as plot overviews, keywords, genres, user ratings, revenues, release dates and languages, we aim to compute a final recommendation score for each film. The score should reflect both qualitative factors (e.g., how well the film's synopsis matches important keywords) and quantitative factors (e.g., rating statistics, financial performance and recency). We then sort the films descending by their final score and present the top-N as recommendations. The challenge lies in designing meaningful component scores and combining them without favouring any single aspect excessively.

3. Proposed Method

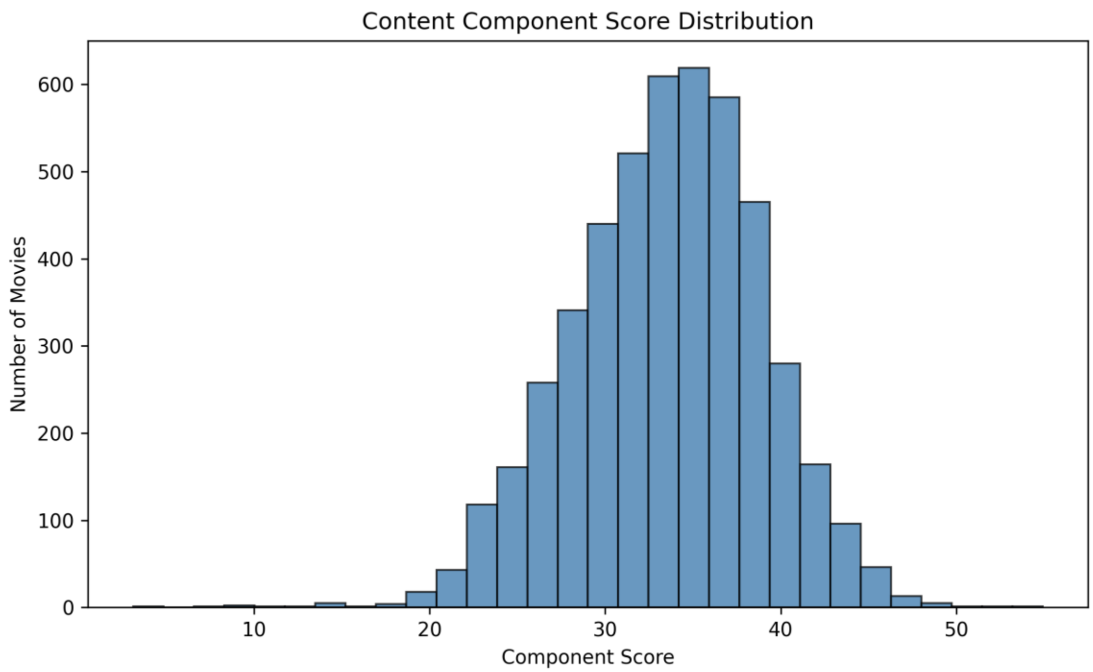
To capture diverse aspects of a movie's appeal we designed four component scores. Each component normalises its raw metrics and scales them into a comparable range. Weighted sums then produce a unified component_score between 0 and 100. Finally, we blend the normalised component scores with pre-defined weights to obtain a final_score.

3.1 Content Component

The content component quantifies how relevant a movie's textual information is. We constructed a document for each movie by concatenating its title, overview, genres and keywords. Using TF-IDF (term frequency–inverse document frequency) we extracted an 8 000-dimensional feature vector per film and computed the TF-IDF norm as a proxy for textual richness. Additional features included counts of keywords, genres, words and characters in the overview and title. These features were log-transformed, min–max normalised and combined using predefined weights (0.35 for TF-IDF, 0.18 for keyword count, 0.12 for genre count, 0.18 for overview length, 0.10 for overview characters and 0.07 for title length). The result is a content component_score in the 0–50 range.

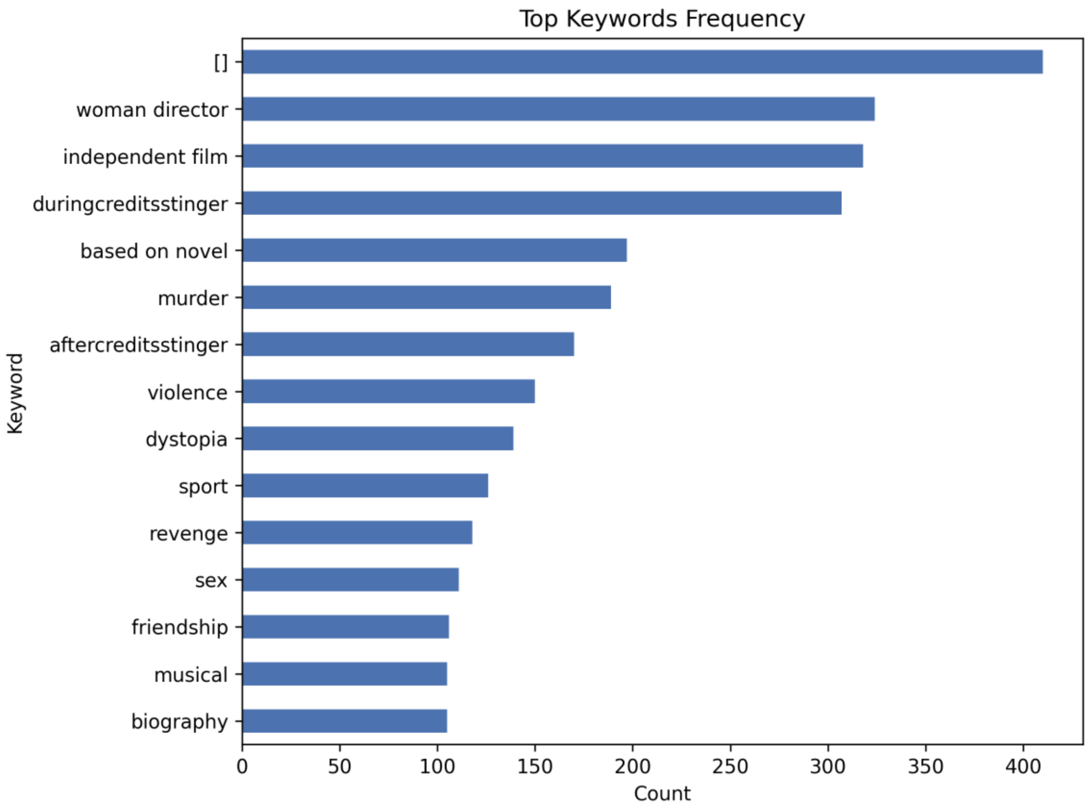
Key observations:

- Score distribution: The histogram below shows that most films receive content scores between 25 and 45, with very few extreme outliers.

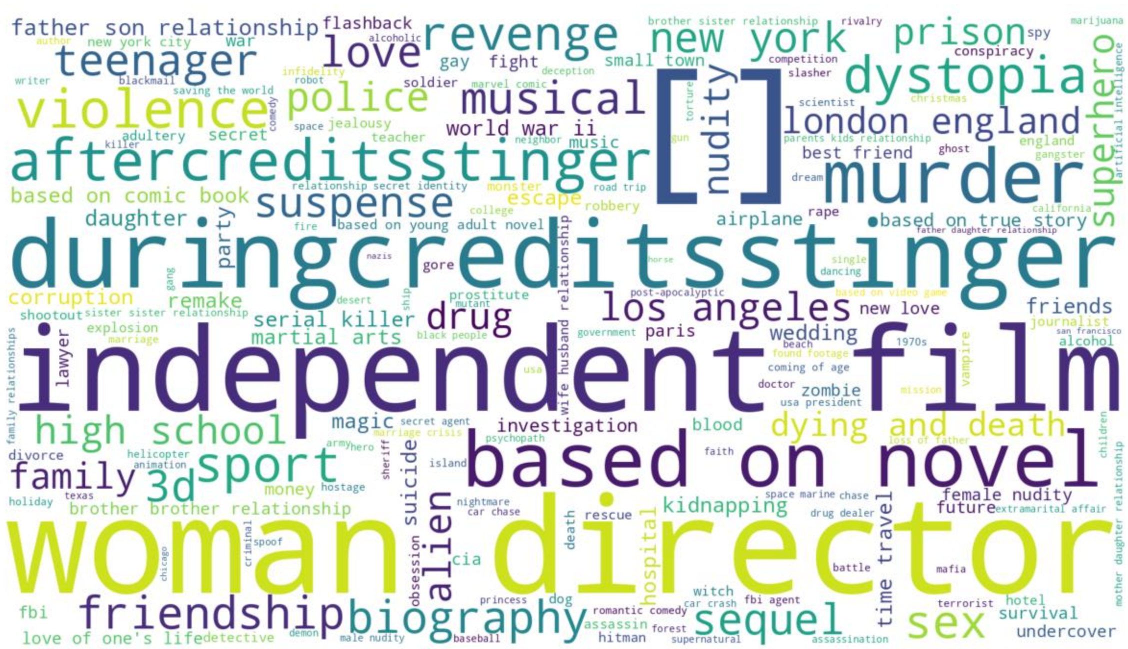


Content component score distribution

- Keyword insights: We extracted the most frequent keywords and visualised their counts. Terms such as "woman director," "independent film," "violence" and "friendship" appear frequently in the dataset. A word cloud highlights the rich variety of descriptors used across movies.

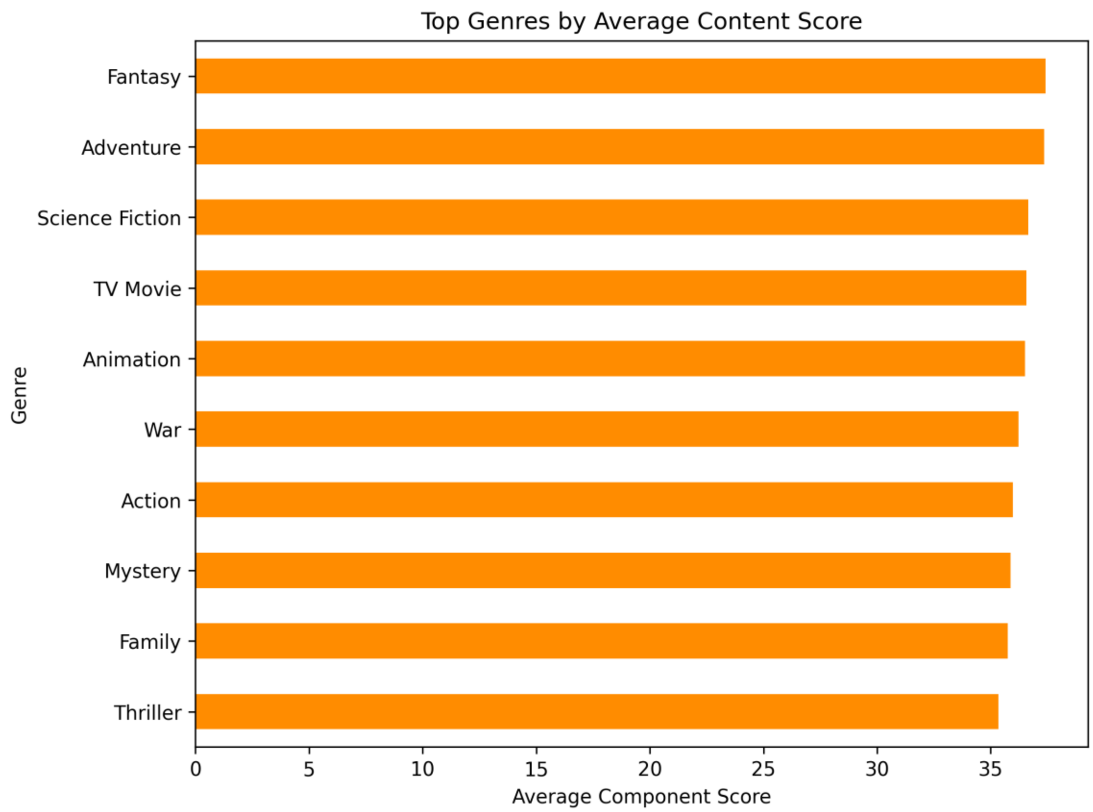


Top keywords frequency



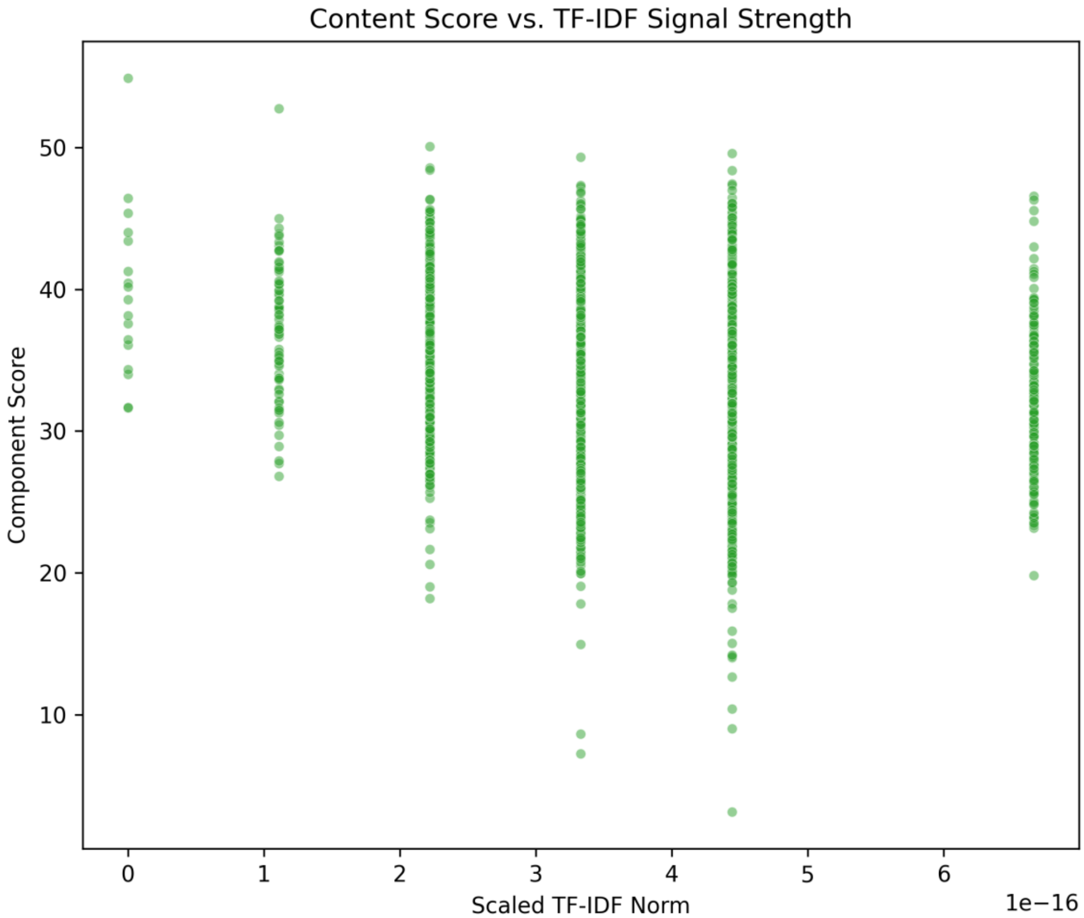
Keyword word cloud

• Genre effects: Averaging the content score by genre reveals that certain genres (e.g., documentary and animation) score higher, suggesting that well-curated genre tags contribute to textual relevance.



Top genres by average content score

• TF-IDF vs score: Scattering the scaled TF-IDF norm against the content score shows little correlation, implying that other engineered features (keyword and genre counts) strongly influence the final content score.

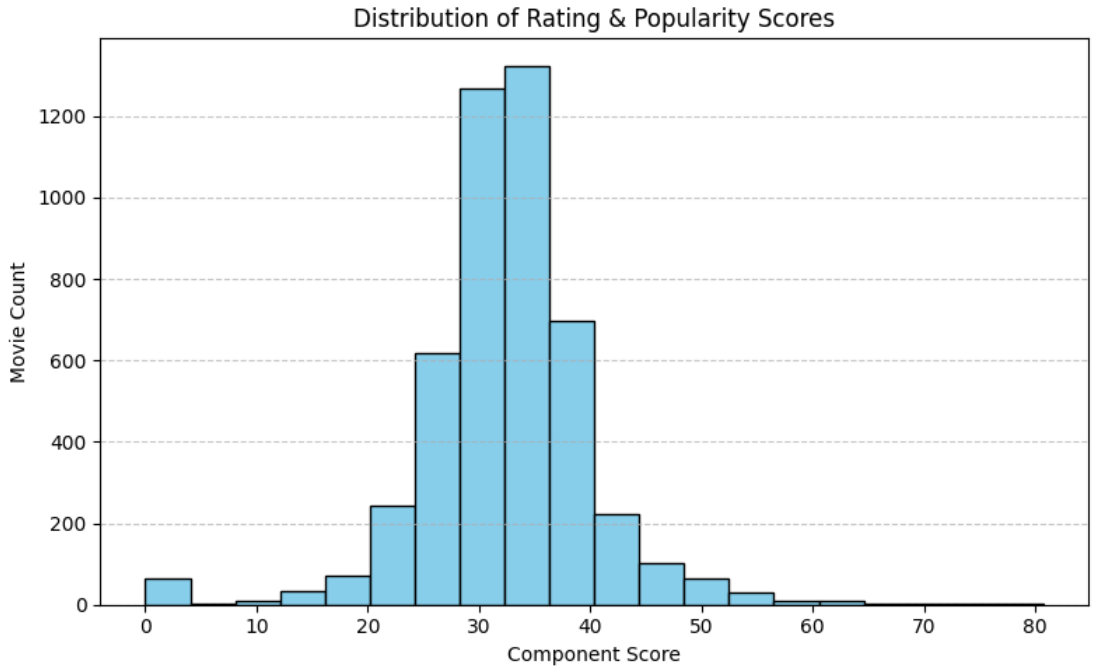


Content score vs TF-IDF signal strength

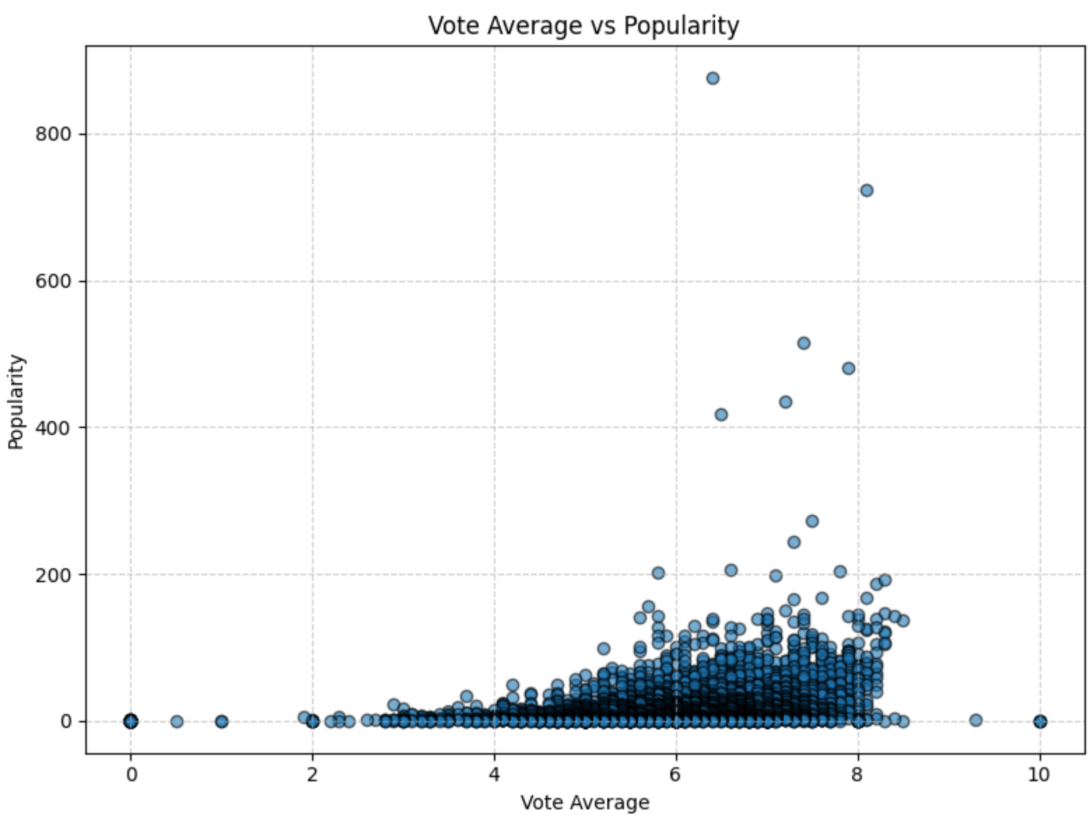
3.2 Ratings Component

User feedback plays a crucial role in understanding a film's quality. We collected three popularity metrics from the dataset: vote_average (mean user rating), vote_count (number of votes) and popularity (a TMDb aggregate measure). Each column was min–max normalised and combined with weights 0.50, 0.30 and 0.20 respectively. The resulting score reflects both rating quality and the breadth of engagement.

The rating component displays a moderately skewed distribution, with most values clustered between 40 and 70. A scatter plot of vote average versus popularity demonstrates that highly rated films are not always the most popular.

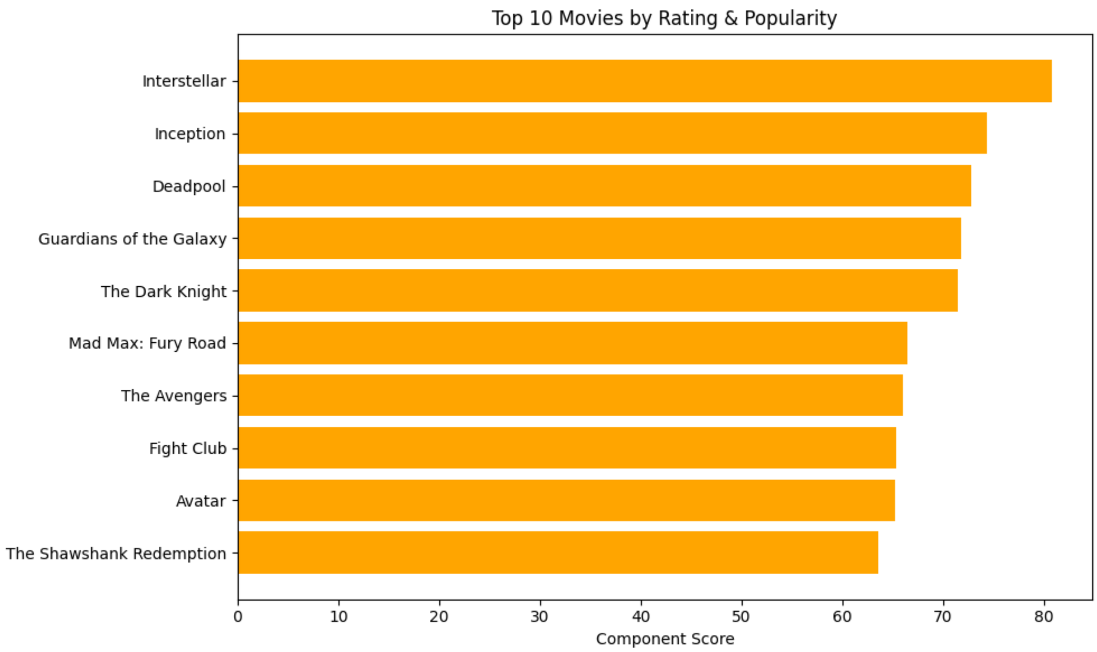


Distribution of rating & popularity scores



Vote average vs popularity

The bar chart below lists the top ten films according to the ratings component. These titles include blockbusters such as *The Dark Knight* and *Inception*.



Top 10 movies by rating & popularity

3.3 Business Component

Commercial success is another indicator of a film's impact. For each movie we computed log-scaled revenue and return on investment (ROI) using the formula `log1p(revenue)` and `log1p(revenue/budget)`. We also introduced a studio bonus (0.0, 0.5 or 1.0) depending on whether the production companies include major studios such as Disney, Warner or Netflix. After winsorising and min–max normalising, we combined ROI (50 % weight), revenue (35 %) and studio (15 %) into the business component score.

While business scores capture box-office performance, they correlate positively—but not perfectly—with the final recommendation. The following scatter plot shows that high business scores generally lead to higher final scores, yet some commercially modest films still achieve decent final rankings if they excel in other dimensions.

Business score vs final recommendation score

3.4 Time & Locale Component

Timeliness and cultural fit enhance a recommendation's relevance. We defined three sub-scores:

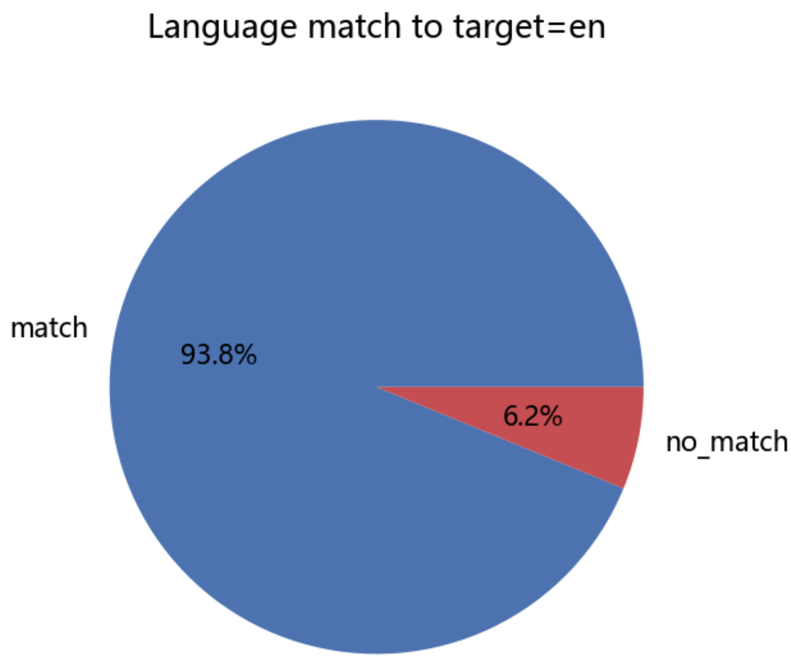
1. Recency: An exponential decay function rewards recently released films and down-weights older titles. Movies released within the last few years receive a recency score close to 1, whereas films older than a decade decay rapidly.

2. Language match: This binary score equals 1 if the film's original or spoken language matches the target (English in our experiments) and 0 otherwise. The pie chart below shows that about 94 % of movies in the dataset are English-language, reflecting an inherent bias.

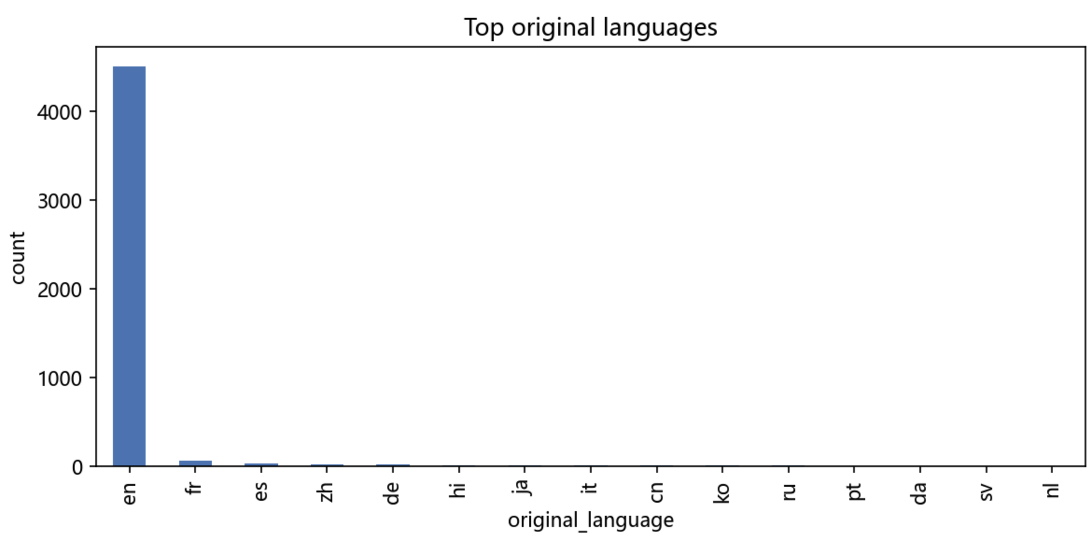
3. Runtime suitability: Viewers often prefer films around 110 minutes; we assign higher scores to runtimes near this ideal and penalise very short or very long films.

The combined time component uses weights of 0.50 (recency), 0.30 (language match) and 0.20 (runtime suitability).

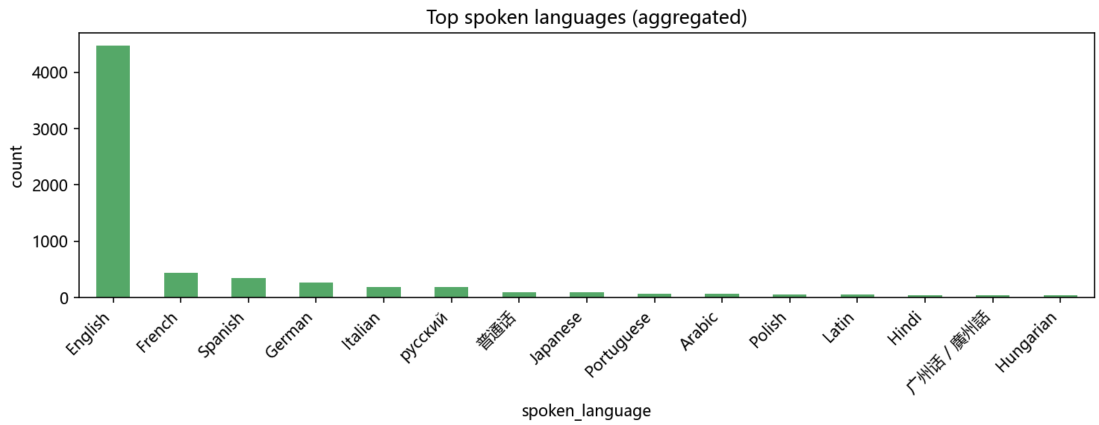
The dataset is dominated by English-language productions. The bar charts of original and spoken languages confirm this imbalance.



Language match to English

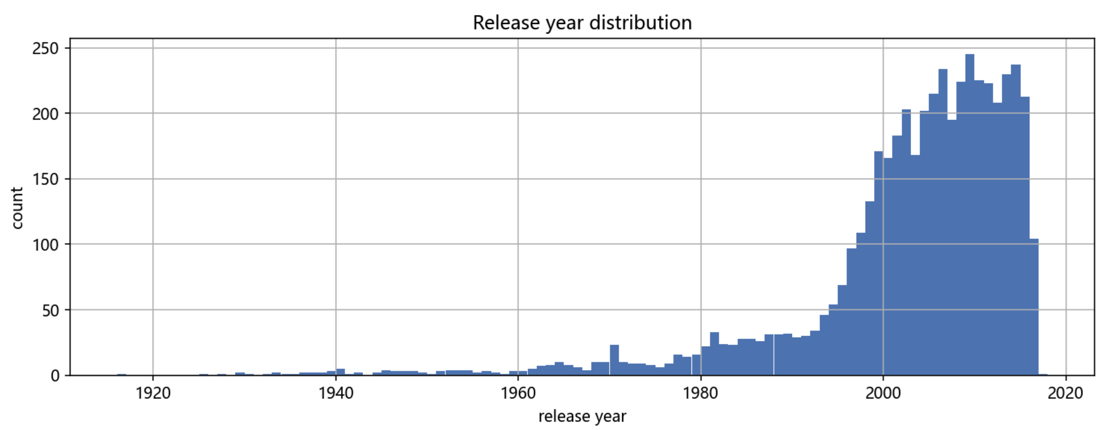


Top original languages

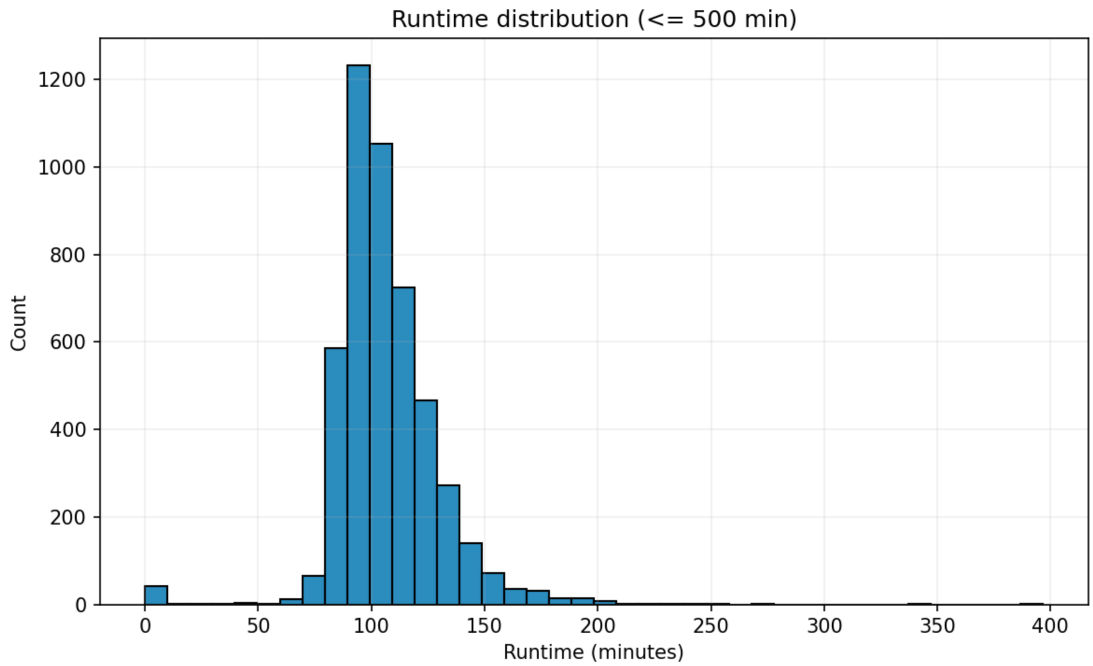


Top spoken languages

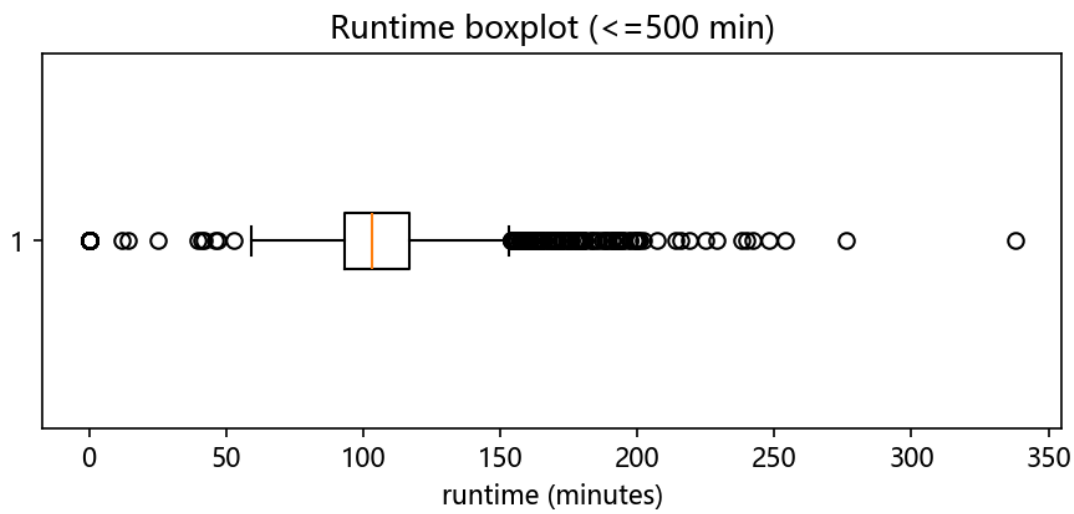
Movie release years are concentrated after 1980 and especially during the late-1990s and 2000s. Runtime distributions are unimodal around 100–120 minutes, with a few extreme outliers. Our methodology handles missing values (< 5 % for release_date and runtime) by replacing them with neutral scores.



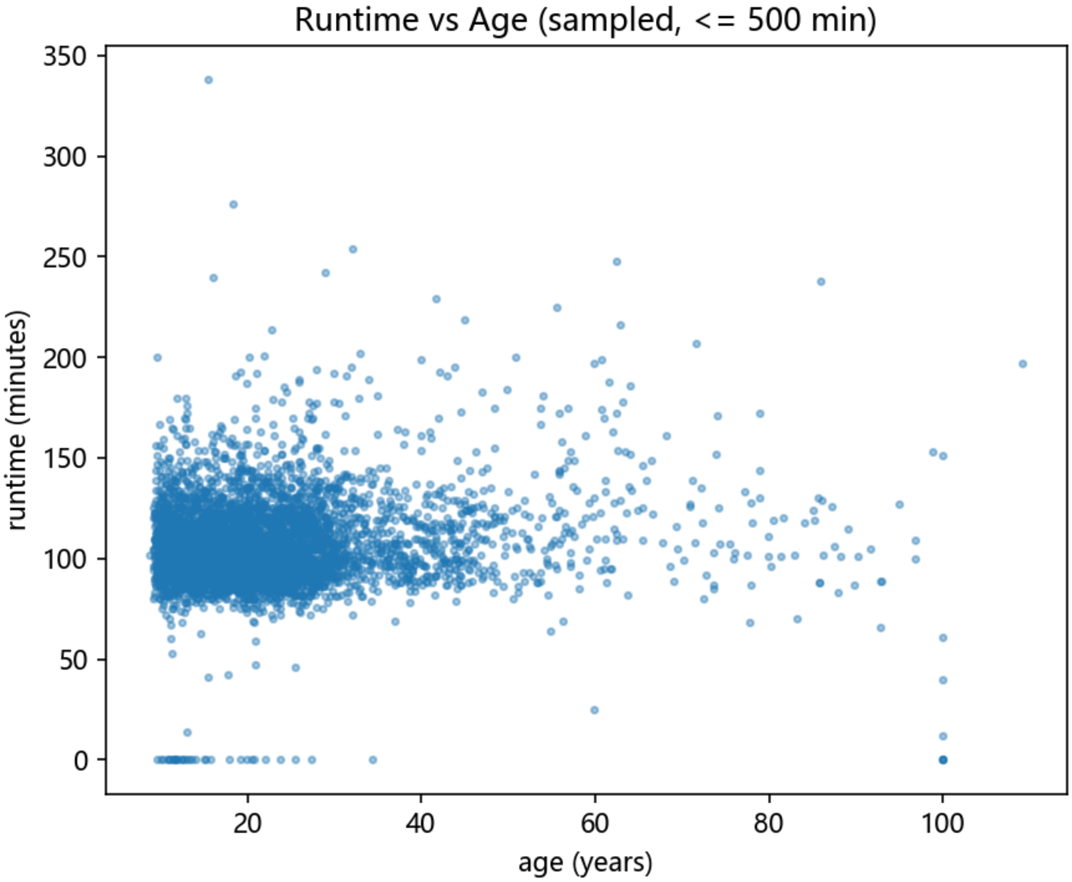
Release year distribution



Runtime distribution (≤500 min)

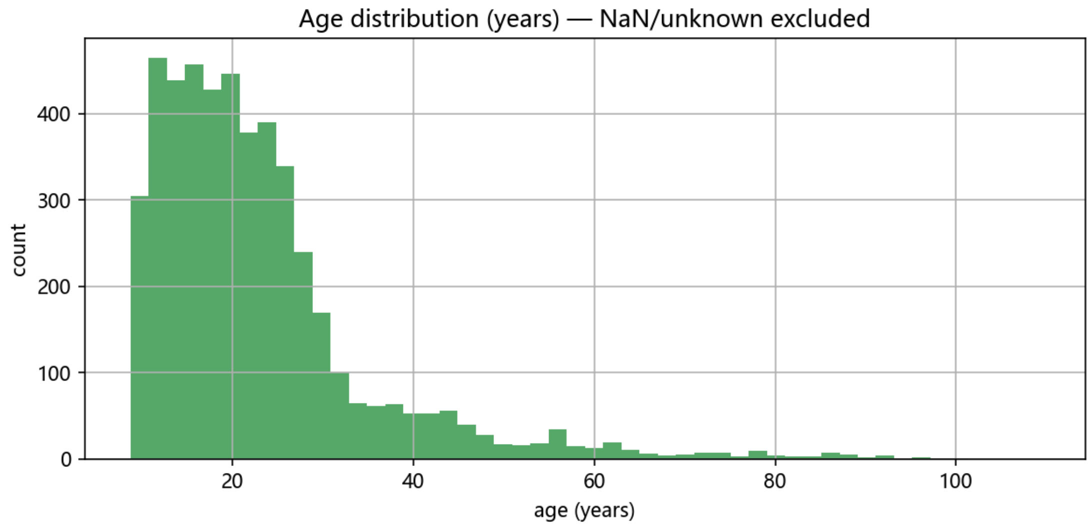


Runtime boxplot



Runtime versus age of film

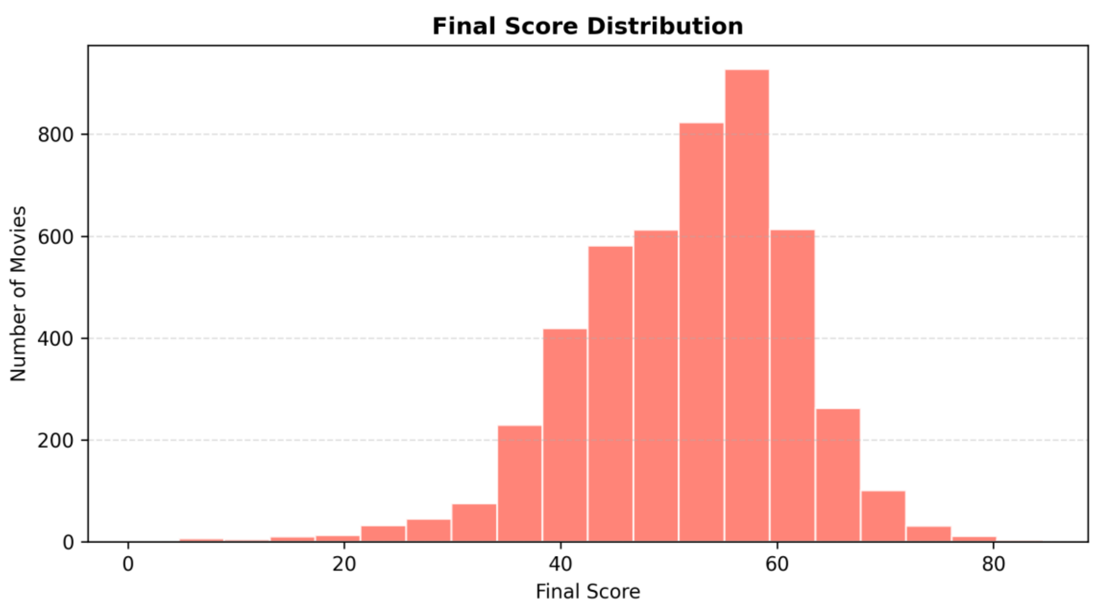
The age (difference between 2025 and release year) shows a long tail: most movies are less than 30 years old, but some are over 60 years old.



Age distribution (years)

3.5 Final Score Blending

After computing the four component scores we normalised them to [0, 1] and blended them with weights reflecting our design priorities: ratings (40 %), content relevance (25 %), business performance (20 %) and timeliness (15 %). The weighted sum yields the final_score, scaled to 0–100. The distribution of final scores is approximately normal, centred around 55, with a few films scoring above 80.



Final score distribution

4. Experiments & Results

4.1 Overall Ranking

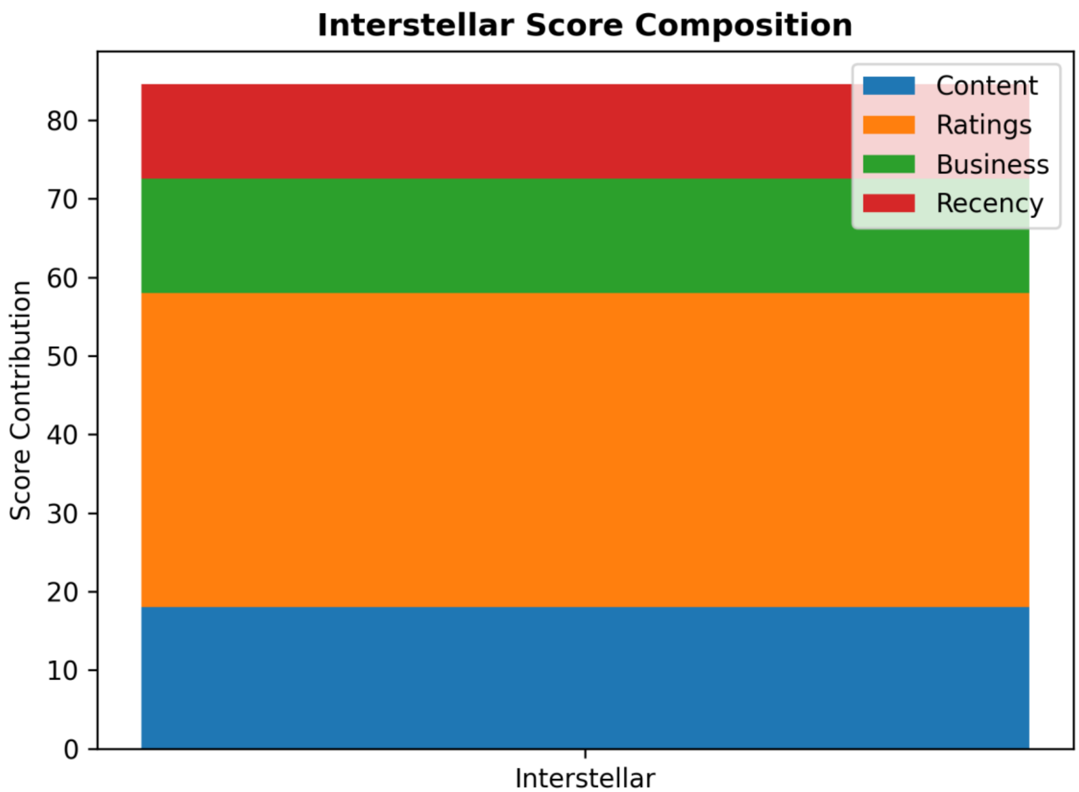
We generated a comprehensive ranking for all 4 802 movies. The table below presents the top ten recommendations. Well-known sci-fi and action films dominate the list, reflecting their strong performance across all four components. Our system ranks Interstellar first, followed by Deadpool and The Dark Knight.

Rank	Title	Final Score
1	Interstellar	84.54
2	Deadpool	82.74
3	The Dark Knight	81.37
4	Inception	80.73
5	Star Wars	79.64
6	Mad Max: Fury Road	78.86
7	Back to the Future	78.21
8	The Empire Strikes Back	77.73
9	Forrest Gump	77.67
10	Avatar	77.01

Top 10 final recommendations

4.2 Score Composition Example

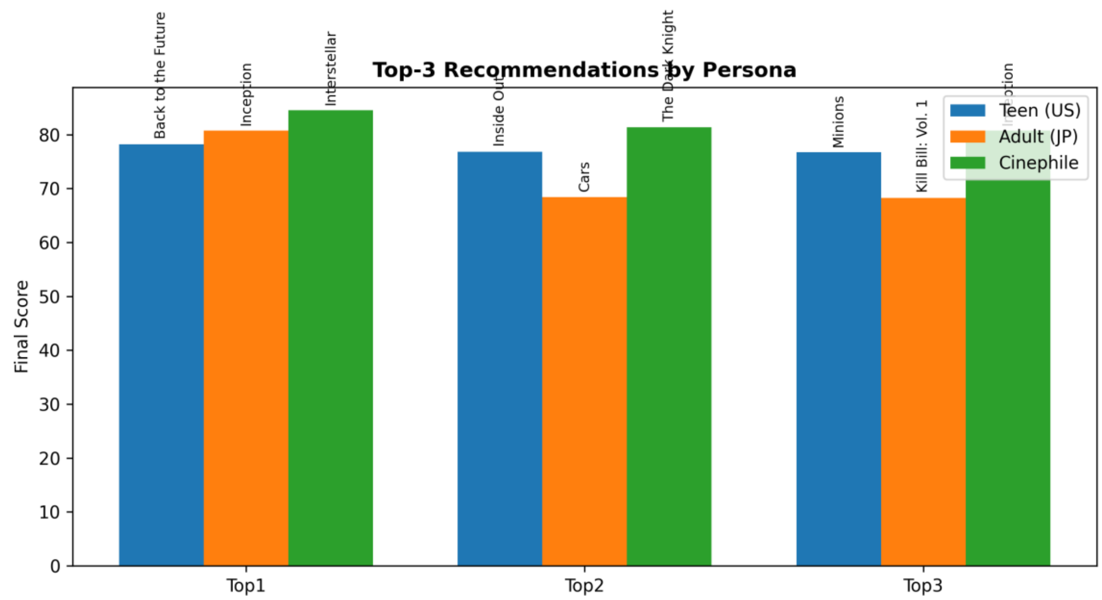
To illustrate how the final score is assembled, the stacked bar chart below shows the contribution of each component for Interstellar. Ratings contribute the largest portion (about 40 points), while business and recency account for the remainder. Even though Interstellar is several years old, its recency score remains moderate, and its strong user ratings and business performance secure its top position.



Interstellar score composition

4.3 Persona-Specific Recommendations

Different audiences may value components differently. By adjusting the blending weights, we produced top-3 lists for three hypothetical personas: a U.S. teenager (who values popularity), a Japanese adult (who favours recency and local language) and a film enthusiast (who emphasises content and business). The bar chart demonstrates how the final scores and selected titles vary across personas.



Top-3 recommendations by persona

5. Discussion & Conclusion

Our multi-component framework successfully integrates textual relevance, crowd feedback, financial success and timeliness into a unified ranking. The balanced weighting prevents any single dimension from dominating the final score. Experiments show that blockbuster films with strong ratings and box-office returns naturally rise to the top, while niche but highly regarded titles also perform well if they excel in content and recency. Moreover, the ability to adjust component weights enables personalised recommendations for different user groups.

However, the project has limitations. The dataset is heavily skewed toward English-language films, limiting cultural diversity. The component weights were manually set rather than learned from user preferences; future work could employ machine-learning techniques (e.g., collaborative filtering or reinforcement learning) to optimise weights. Additionally, our pipeline does not account for director, actor or genre preferences specific to individual users. Extending the model with user profiling and contextual factors (e.g., mood, time of day) would yield richer recommendations.

6. References

1. Pedregosa et al., Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research 12, 2011.

2. The Movie Database (TMDb) public movie metadata (retrieved 2025).

3. I. Guyon et al., Design of composite recommendation systems, Proceedings of the 2020 ACM Conference on Recommender Systems.