THE WATCHLIST

Movie Recommendation Analytics Project:

Building a Personalized Cross-Platform Recommendation System from Multi-Source Movie Data

Project By: Parsa Hassassedighi

The 400 Blows (1959 - François Truffaut)

Table of Contents

Introduction

1. Project summary

This project builds a personalized movie recommendation system from a user's watchlist, ratings, and external movie metadata. The workflow combines automated data fetching, enrichment, cleaning, transformation, segmentation, and ranking to recommend films from a 50,000-title candidate pool. Instead of relying on one platform's suggestions, the system is designed to support broader discovery, explain user taste across multiple dimensions, and surface stronger recommendations using both preferences fit and cinematic-quality signals.

---

2. Problem statement

Most movie recommendation systems are platform-specific, trend-driven, and optimized for broad engagement rather than precise discovery. This creates three major issues: limited cross-platform discovery, underexposure of niche and independent films, and filter-bubble behavior that reduces exploration. This project was designed to address those gaps with a broader, more explainable, and more personalized recommendation workflow.

Dimension	Standard Platform Recommender	The WATCHLIST
Cross-platform discovery	Limited	Supported
Long-tail / niche discovery	Often weak	Stronger focus
Explainability	Low	Higher
Taste modeling	Broad / generalized	Multi-aspect + stream-based

Table 1: Featuresof the project in comparison with main-stream streaming platforms

---

3. Objective

The objective was to design an end-to-end recommendation analytics workflow that:

- models user taste from watchlist and ratings
- integrates external movie data from multiple sources
- transforms raw records into clean, ready-to-use analytical inputs
- ranks candidates from a large pool using both preference and quality signals
- improves recommendation relevance without collapsing into popularity-only or trend-based logic

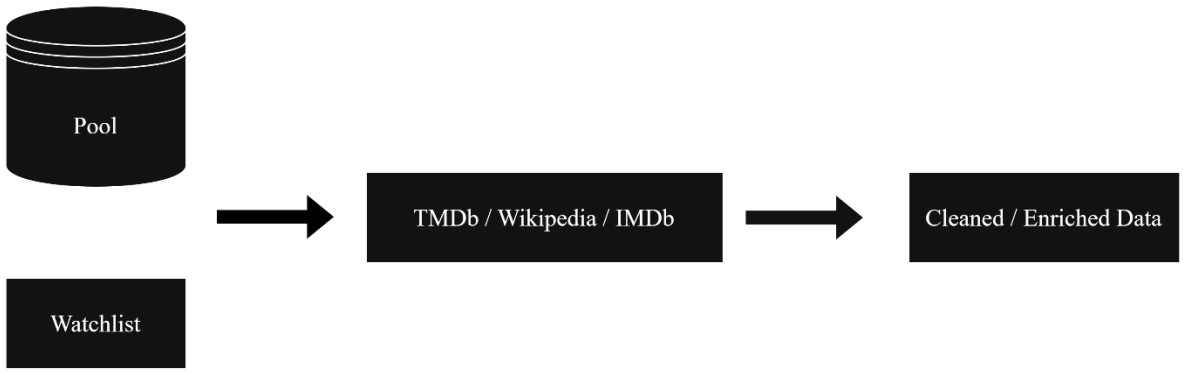
Data Engineering

4. Data sources and collection

The project uses multiple sources:

- personal watchlist and rating data
- TMDb metadata
- Wikipedia plot summaries, including multilingual fallback
- IMDb rating and vote count data
- a 50,000-title candidate pool for large-scale scoring

A major part of the project was building an **automated multi-source data pipeline** that fetches, enriches, validates, and standardizes movie records into clean, reusable datasets for downstream analysis and recommendation scoring.



---

5. Automated data pipeline, cleaning, and enrichment

Rather than manually cleaning a static dataset, the project uses an automated workflow to retrieve and prepare movie data. The pipeline handles:

- data fetching from external APIs and datasets
- missing and weak text recovery
- multilingual plot retrieval and translation
- schema alignment across sources
- null handling and datatype repair
- checkpointing and resumable processing
- row-count and ID-integrity validation

To scale enrichment across the 50k pool, the workflow uses a two-pass design:

- Pass 1: fast TMDb-only enrichment
- Pass 2: Wikipedia enrichment only for low-confidence rows

Stage	Purpose	Scope	Main Inputs	Main Outputs
Pass 1	Fast large-scale TMDb enrichment	All rows	TMDb IDs	Core metadata + TMDb-rich text
Pass 2	Selective quality improvement using Wikipedia	Low-confidence rows only	Pass 1 parquet + wiki fetch logic	Enriched plot text + improved final text

Table 2: Two-Pass Data Enrichment Pipeline

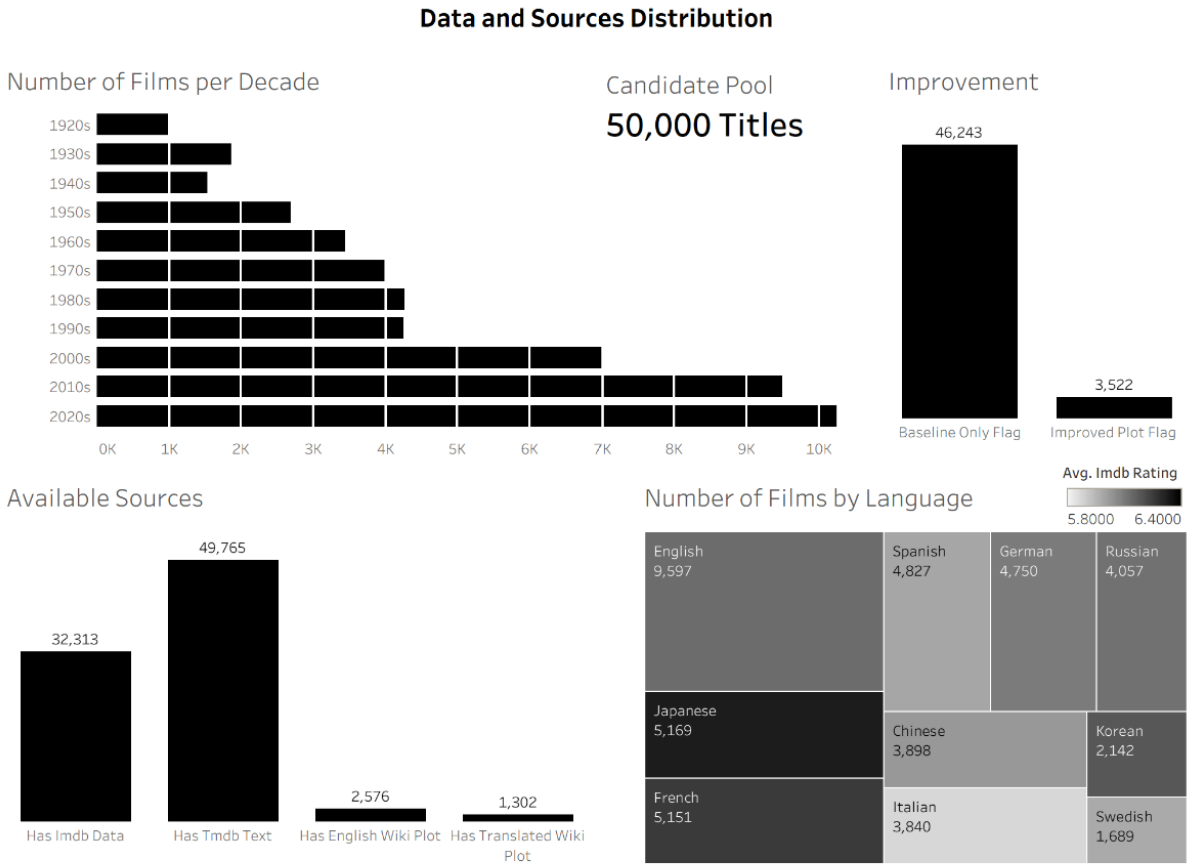


Figure 1: Source Coverage Dashboard

6. Feature engineering and transformation

Raw metadata is transformed into model-ready features. Key engineered features include:

- rich movie text fields
- four aspect-specific text channels:
    - theme
    - mood
    - style
    - context
- semantic embeddings for each aspect
- keyword and descriptor features
- centered rating-based user preference vectors
- overlapping taste streams derived from clustering

This step converts raw movie records into structured analytical representations suitable for segmentation and ranking.

Title	text_theme	text_mood	text_style	text_context
Cinema Paradiso	title: Cinema Paradiso genres: Drama, Romance themes: parent child relationship, sicily, italy, movie business, kiss, coming of age, flashback, magic realism, haunted by the past, censorship, nuovo cinema paradiso, movie theater, comforting, compassionate, italy movie, asks alfredo, theater comforting …	title: Cinema Paradiso mood: nostalgic overview tone: A filmmaker recalls his childhood, when he fell in love with the movies at his village's theater and formed a deep friendship with the theater's projectionist.	title: Cinema Paradiso genres: Drama, Romance	title: Cinema Paradiso language: it year: 1988 decade: 1980s genres: Drama, Romance overview context: A filmmaker recalls his childhood, when he fell in love with the movies at his village's theater and formed a deep friendship with the theater's projectionist.

Table 3: Example of Aspect-Specific Text Representation for a Single Movie

Machine Learning

7. Preference modeling and segmentation

The recommendation logic models preference at two levels.

First, it builds user-level taste vectors from ratings to represent:

- overall taste
- positive taste
- negative taste

Second, it segments the watchlist into soft overlapping “taste streams” so that the system can capture multiple lanes of preference instead of forcing a single taste profile.

This makes the recommendation logic more interpretable and better suited for exploration.

stream_id	stream_strength	n_rated_movies	avg_membership	max_membership	weighted_avg_rating
0	2.256046101	266	0.500023031	0.570564665	7.302357884
1	-0.41830174	266	0.499976969	0.573427337	7.281776507

Table 4: Taste Stream Summary Metrics

Table 2 shows the main statistics for each inferred taste stream. “stream_strength” indicates how strongly that stream aligns with the user’s ratings, while “weighted_avg_rating” shows the average rating level within that stream. “avg_membership” and “max_membership” describe how strongly movies are associated with each stream.

	Stream_id 1	Stream_id 2
representative_titles	Cries And Whispers Autumn Sonata Mother! Howl's Moving Castle Nostalgia	Scarface The Departed Public Enemies John Wick Pulp Fiction

Table 5: Representative Movies for Each Taste Stream

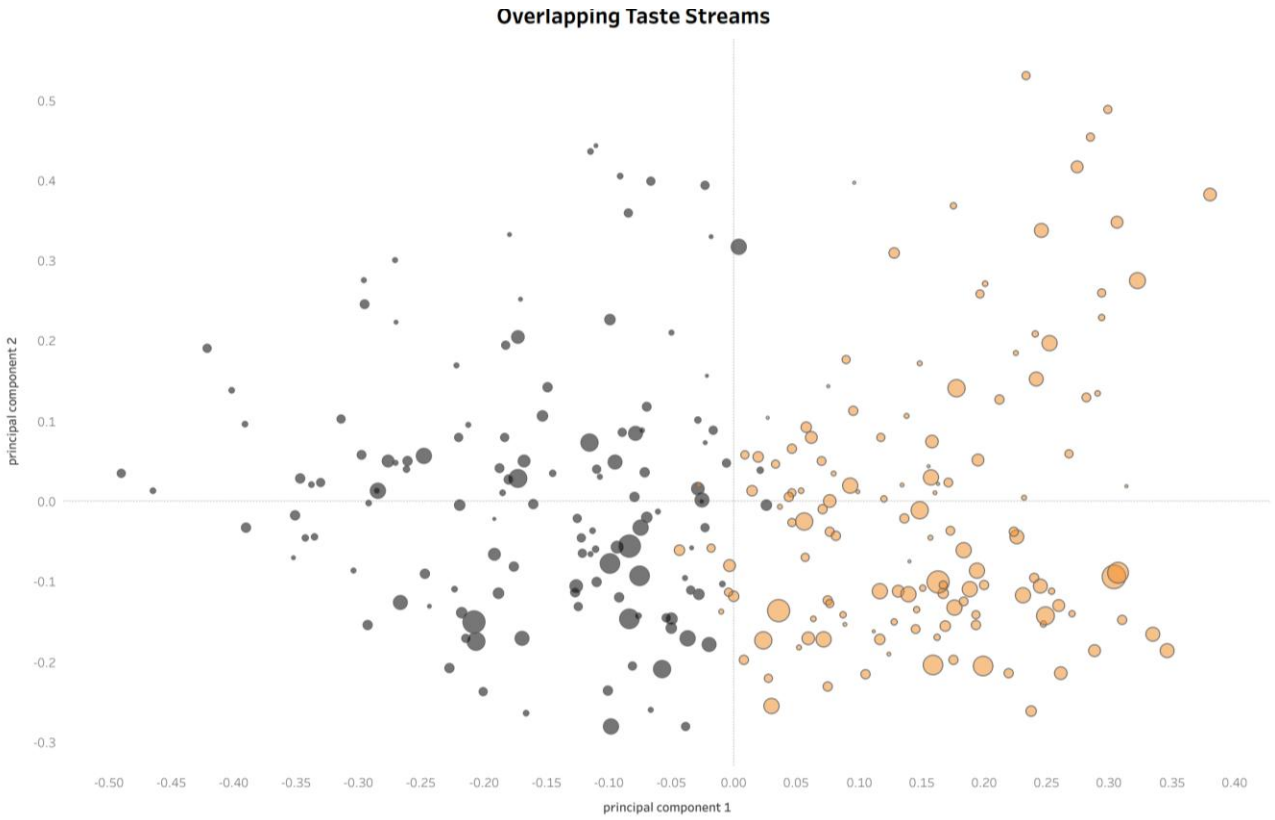


Figure 2: Rated Watchlist Movies in 2D PCA Space by Taste Stream

This chart shows rated watchlist movies projected from high-dimensional embedding space into two PCA dimensions for visualization. Each point is a movie, and color indicates its primary taste stream. The purpose is to show that user preference is not concentrated in one single cluster: the watchlist contains multiple taste regions, with some movies sitting clearly inside one stream and others overlapping between streams. PC1 and PC2 are mathematical projection axes, so they should be interpreted as summary dimensions of variation rather than literal features like genre or mood. The chart is useful because it makes the stream structure visible and supports the decision to model taste as overlapping preference lanes rather than one flat profile.

---

8. Recommendation scoring

Candidate movies from the 50,000-title pool are ranked using a weighted scoring framework that combines:

- aspect-level similarity
- stream alignment
- negative-preference penalties
- IMDb-based cinematic-quality signals
- metadata richness and text reliability

This creates a recommendation workflow that is not only taste-aware, but also more selective about surfacing worthwhile films.

Score component	What it measures	Why it matters
Taste similarity	How well a movie matches the user's learned preferences	Keeps recommendations personalized
IMDb quality score	General audience/critical approval signal	Helps avoid weak matches with poor overall reception
Diversity or novelty term	How different the movie is from already selected items	Prevents recommendations from becoming too repetitive

Table 6: Recommendation Scoring Components

---

9. Validation and iterative refinement

The project was developed iteratively using validation and audit checks throughout the workflow. These included:

- row and ID integrity checks
- enrichment coverage auditing
- confidence tracking before and after enrichment
- manual review of suspicious Wikipedia matches
- refinement of clustering logic when silhouette favored overly simple splits
- ranking adjustments when outputs were too broad or low-quality

This made the project closer to a real analytics cycle than a one-pass model build.

Results and Conclusion

10. Sample results

The final output of the project is a ranked set of movie recommendations drawn from a 50,000-title candidate pool. Recommendations are first scored using the user's learned preference structure, including aspect-level similarity, soft stream alignment, and negative-preference penalties. A second ranking layer then incorporates cinematic-quality signals using IMDb-based weighted ratings, metadata richness, and text reliability. This creates a more selective recommendation output that favors both preference fit and stronger film quality.

Rank	Title	Score V1	V1 Rank	IMDb Weighted Rating	IMDb Quality Score	Final Score
1	A Short Film About Love	0.281760848	74	8.05	0.79099538	0.409069481
2	Bad Poems	0.291671352	27	7.23	0.709960165	0.396243555
3	Simple Things	0.322726048	2	6.47	0.611376297	0.39488861
4	And the Woman Shall Fear Her Husband	0.273721952	100+	7.46	0.744914875	0.391520183
5	World Gone Mad	0.247434927	100+	8.24	0.820452605	0.390689346
6	Erotic Stories	0.321336858	3	6.17	0.593532531	0.389385777
7	Clownery	0.304927947	12	6.27	0.641505137	0.389072245
8	Sexomania	0.315522826	6	6.03	0.608234312	0.388700698
9	Sequence of Feelings	0.315763654	5	5.97	0.603647823	0.387734696
10	Life Feels Good	0.273127779	100+	7.34	0.726396553	0.386444972

Table 7: Top Recommendation Examples After Quality Re-Ranking

This table shows sample output from the final ranking system. "Score V1" is the original taste-based score, "IMDb Weighted Rating" is the confidence-adjusted quality signal, and "Final Score" is the combined result after adding the cinematic-value layer. The table demonstrates how the system moves from pure taste matching to stronger, more selective recommendations.

---

11. Key outcomes

- Built an automated multi-source data workflow that fetches, enriches, validates, and standardizes movie records into analysis-ready datasets.
- Created a 50,000-title candidate pool with TMDb, Wikipedia, and IMDb-derived features.
- Engineered multi-aspect representations of movies using theme, mood, style, and context text channels.

- Modeled user preference through:
  - centered rating-based taste vectors
  - positive and negative preference signals
  - overlapping taste streams
- Developed a recommendation scoring framework that combines:
  - relevance to user taste
  - stream-based preference alignment
  - cinematic-quality evaluation
- Produced a strong baseline recommender and established a foundation for:
  - multi-query retrieval
  - diversity-aware reranking
  - larger-scale indexing in later versions

---

12. Future improvements

Planned next steps:

- multi-query retrieval
- diversity and exploration-aware reranking
- stronger reliability filtering for noisy enrichment
- ANN indexing for larger-scale retrieval
- stronger recommendation explanations

Cinema Paradiso (1988 - Giuseppe Tornatore)