International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Movie Bubble: A Group-Centric Movie Recommendation System

SK. Mujafar Ahmed[1], K. Yashwanth Reddy[2], B. Sai Charan[3], V. Vineeth[4], V. Vineel[5]
[1]Assistant Professor, [2–5]UG Scholar
Department of Artificial Intelligence & Machine Learning, School of Engineering,
Malla Reddy University (MRUH), Hyderabad, India

***Abstract* - Choosing a movie to watch together is a common yet contentious challenge for groups of friendsand family. Traditional recommendation systems are architected for individual users and fail to account forthe complex dynamics that emerge when multiple people with dif ering tastes, moods, and streamingplatformsubscriptions attempt to reach a consensus. This paper presents Movie Bubble, a group-centric movierecommendation system that bridges this gap by combining individual user profiling, hybridfilteringtechniques (collaborative and content-based), mood-based NLP filtering, and democratic group consensusaggregation strategies. The system organises users into temporary groups called bubbles, aggregates theirpreferences, and generates a fair, inclusive shortlist of movie suggestions. A built-in polling mechanismempowers all group members to vote, ensuring participatory decision-making. Consensus strategies includingaverage aggregation, least misery, and Borda count voting are implemented as backend logic to maximisecollective satisfaction. The system integrates with the TMDb API and TMDB Watch Providers API for up-to-date movie data and streaming availability, and uses PostgreSQL as its sole persistence layer. Resultsdemonstrate that Movie Bubble significantly improves the group movie-selection experience, transformingit from a source of conflict into a collaborative and enjoyable activity.**

Keywords: Group Recommender System, Collaborative Filtering, Content-Based Filtering, SBERT, Mood-Based Filtering, Borda Count, Consensus Aggregation, Streaming Platforms, TMDb API, PostgreSQL

1. INTRODUCTION

The proliferation of streaming platforms such as Netflix, Amazon Prime Video, Disney+, and HBOMaxhasdramatically expanded the catalog of available movies. While this benefits individual viewers, it paradoxically exacerbates decision fatigue when groups attempt to choose what to watch together. Asingleuser benefits from a personalised recommendation engine. However, when a group of four or five peopleeachbring distinct genre preferences, mood inclinations, and platform subscriptions, no single individual recommendation engine serves the group adequately.

This challenge—the "what should we watch tonight" problem—is deceptively difficult computationally. It involves multi-stakeholder optimisation, fairness constraints, mood-aware filtering, and real-time platformavailability filtering. A recommendation of a critically acclaimed horror film satisfies some members whilealienating others who prefer romantic comedies. Similarly, a film unavailable on a member's subscriptionisimpractical regardless of how well it matches aggregate taste.

Movie Bubble addresses this gap. The system uses the metaphor of a bubble—a temporary group session—where users contribute individual profiles and collectively arrive at a movie choice through a transparent, fairprocess. Contributions of this work include: (1) a hybrid recommendation engine combining collaborativeandcontent-based filtering; (2) a mood-based NLP filtering layer using SBERT semantic embeddings; (3)
multiple group consensus aggregation strategies as backend logic; (4) an interactive polling interface; (5)integration with the TMDb API and TMDB Watch Providers API; and (6) a clean, scalable architecturebacked entirely by PostgreSQL.

2. LITERATURE REVIEW

2.1 Individual Recommendation Systems

The foundation of modern recommender systems was laid by collaborative filtering (CF), popularisedbytheGroupLens project in the mid-1990s. CF operates on the principle that users who agreed in the past will agreein the future [5]. Content-based filtering (CBF) analyses item attributes such as genre, director, cast, andplot keywords to recommend items similar to those a user has previously enjoyed [8]. Hybrid systems combiningCF and CBF have demonstrated superior performance over either approach

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

alone, mitigating the cold-start problem and rating matrix sparsity [3].

2.2 Group Recommendation Systems

Group recommendation is a well-recognised sub-field. Early work by O'Connor et al. [7] introducedPolyLens, one of the first group recommender systems. Research has explored aggregation strategies intwofamilies: Aggregation of Recommendations (AR) and Aggregation of Preferences (AP). Masthoff [6] notesthat least misery produces more equitable outcomes for heterogeneous groups. The Borda count methodprovides a rank-based mechanism effective in group settings [2].

2.3 Fairness and Interactivity

Quintarelli et al. [9] highlight that users are more satisfied with group recommendations when theyperceivethe decision process as fair, regardless of whether the final choice was their top preference. This motivatesthepolling and voting mechanism in Movie Bubble.

2.4 Mood-Aware Recommendation

Adomavicius and Tuzhilin [1] demonstrated that incorporating situational context—including emotional state—substantially improves recommendation relevance. Reimers and Gurevych [10] introducedSBERT, enabling fine-grained semantic matching between natural language mood descriptions and movie overviews, providing a practical pathway for mood-aware filtering.

2.5 Streaming Platform Integration

The TMDB Watch Providers API provides a reliable consolidated source for streaming availabilitydatabyregion [11], enabling recommendation pipelines to filter candidates to only practically accessible titles. Thisisa largely unexplored constraint in prior group recommendation literature.

3. METHODOLOGY

3.1 System Architecture Overview

Movie Bubble is a full-stack web application with a modular backend recommendation engine. Thearchitecture consists of five primary components: (1) User Profile Management, (2) Bubble GroupSessionManagement, (3) Hybrid Recommendation Engine, (4) Group Consensus Aggregation Module, and(5)
Polling and Voting Interface. These communicate through RESTful APIs and are backed by a PostgreSQLpersistence layer.

3.2 User Profiling

Each user maintains a profile capturing: genre preferences as weighted vectors; streaming platformsubscriptions; liked movies stored as binary preferences used to infer taste patterns for collaborativeandcontent-based filtering; and mood preferences indicating preferred emotional categories. Profiles areinitialised through an onboarding questionnaire and updated via ongoing interactions, all storedinPostgreSQL.

3.3 Bubble Formation

A bubble is a temporary group session created by a host user who invites others. Each member's profileispulled from the database upon joining. The bubble's target streaming platforms are derived fromtheintersection or union of members' subscriptions. All bubble metadata, session state, and vote recordsarepersisted in PostgreSQL.

3.4 Hybrid Recommendation Engine

The engine operates in a multi-stage pipeline. Individual Scoring: Collaborative Filtering (CF) applies SVD-based matrix factorisation to a user-item preference matrix derived from liked movies. Content-BasedFiltering (CBF) encodes movie attributes using TF-IDF vectors and computes cosine similarity withuserpreference vectors. Hybrid Scoring combines both: $final\_score = \alpha \times CF\_score + (1-\alpha) \times CBF\_score$, where$\alpha$ is tuned based on data availability.

Platform Filtering: Candidate movies are filtered through the TMDB Watch Providers API to retainonlythose available on at least one of the bubble's target streaming platforms.

3.4.1 Mood-Based NLP Filtering

Following platform filtering, the pipeline applies mood-based semantic filtering using seven mood categories(Table 1). The group

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

selects a mood for the session. Each mood category is encoded as a dense semanticvector using SBERT (Sentence-BERT). Each candidate movie's plot overview is similarly encoded, andcosine similarity is computed between the group mood embedding and each movie's embedding. Moviesscoring below a configurable threshold are filtered out; the remainder are re-ranked by combinedmoodalignment and hybrid recommendation scores.

Table 1: Mood Categories and Descriptions

Mood	Description
Lighthearted	Fun, comedic, feel-good films
Emotional	Moving, dramatic, tear-jerking stories
Action	High-energy, thrilling, fast-paced content
Dark	Gritty, intense, psychologically heavy films
Fantasy	Imaginative, fantastical, world-building narratives
Romantic	Love stories, relationship-driven dramas
Thoughtful	Slow-burn, philosophical, introspective films

3.5 Group Consensus Aggregation

Given the mood-filtered, platform-available candidate list, individual scores are aggregated using one of threebackend consensus strategies: (1) Average Aggregation: group_score(m) = (1/n) × Σ score_i(m)—balancespreferences across all members. (2) Least Misery: group_score(m) = min score_i(m)—ensures no member isdeeply dissatisfied. (3) Borda Count: a movie receives points equal to the number of candidates rankedbelowit by each member; points are summed for a collective ranking robust to extreme preferences. The systemselects the strategy based on group composition heuristics. The top-N shortlist (default N=10) is thenpresented for voting.

3.6 Polling Mechanism

The shortlisted movies are displayed to all bubble members through an interactive polling interface. Eachmember can upvote preferred options. Votes are persisted in real-time to the PostgreSQLdatabaseandreflected on page refresh. The final recommendation is the movie with the highest total votes; ties are brokenby falling back to the Borda count ranking. The live vote distribution is visible to all members, encouragingtransparency and engagement.

3.7 Data Sources and API Integration

Table 2 summarises the external data sources integrated into Movie Bubble.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Table 2: External Data Sources

Source	Purpose
TMDb API	Movie metadata: titles, genres, cast, crew, releaseratings, plot summaries
TMDB Watch Providers	API Real-time streaming availability by region via/movie/{id}/watch/providers endpoint

dates,

3.8 Database Architecture

Movie Bubble uses PostgreSQL as its sole persistence layer, managing all structured relational data includinguser profiles, genre preferences, liked movies, platform subscriptions, mood preferences, bubble sessionmetadata, group membership, poll records, votes, and movie metadata. This single-database approachsimplifies deployment, ensures transactional consistency, and reduces infrastructure complexitywithout sacrificing performance at current scale.

4. RESULTS AND DISCUSSION

4.1 Prototype Implementation

A functional prototype has been developed as a full-stack web application. The backend is built withNode.js, Express, and TypeScript, exposing a RESTful API consumed by the frontend. Adedicatedhypothesis/research module written in Python handles NLP analysis tasks—specifically SBERT-basedmoodfiltering (Section 3.4.1)—and is invoked as a subprocess by the Node.js backend when mood scoringisrequired. All persistent state is managed in PostgreSQL.

4.2 Recommendation Quality

Preliminary evaluation used a subset of the MovieLens 25M dataset. The hybrid model ($\alpha = 0.6$ in favour ofCF) achieved MAE = 0.71 and RMSE = 0.93, competitive with CF-only (MAE: 0.79) and CBF-only(MAE: 0.88) baselines. The hybrid approach demonstrated particular improvement for users with sparse preferencehistories, where CBF compensated for CF's cold-start weakness.

4.3 Group Aggregation Strategy Comparison

A user study was conducted with 12 test groups (3–5 members each). Participants rated satisfactionona5-point Likert scale. Average Aggregation yielded highest satisfaction (mean: 3.9/5) for homogeneous groups. Least Misery was preferred (mean: 4.1/5) in heterogeneous groups where at least one member hadstrongdislikes [6]. Borda Count received the most consistent scores across all group types (mean: 3.8/5), indicatingrobustness as a default strategy. Transparent process was reported as significantly improving perceivedfairness [9].

4.4 Mood-Based Filtering Effectiveness

The SBERT mood filtering layer was evaluated by presenting users with shortlists generated with andwithout mood filtering. Users reported mood-filtered recommendations felt more contextually appropriate in78%ofsessions. Lighthearted and Action were the most frequently selected moods. Average cosine similarityscoresranged from 0.61 (Thoughtful) to 0.74 (Action), reflecting varying semantic alignment betweenmooddescriptions and movie overviews.

4.5 Polling Engagement

All 12 test groups reported that the voting step made them feel more invested in the outcome. Notably, 83%of participants stated they would be satisfied watching the group's final choice even when it was not theirindividual top preference, attributed to the perceived fairness of the process.

4.6 Platform Filtering Effectiveness

TMDB Watch Providers API integration successfully filtered out unavailable movies in 100%of test sessions. On average, platform filtering reduced the candidate pool by 38%, ensuring all final recommendations werepractically actionable. This feature was rated as most practically useful by 9 out of 12 test groups.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

4.7 Limitations

The user study is limited in scale (12 groups); a larger study is needed for statistical significance. SVD-basedCF requires sufficient liked movies per user; new users rely more heavily on CBF. TMDB Watch ProvidersAPI coverage varies by region. The mood filtering threshold is a fixed hyperparameter; adaptive thresholdingwould improve robustness.

5. CONCLUSIONS

This paper presented Movie Bubble, a group-centric movie recommendation systemaddressing thewell-known but underserved problem of collaborative movie selection. By combining a hybrid recommendationengine, a mood-based NLP filtering layer using SBERT semantic embeddings, multiple backendgroupconsensus aggregation strategies, an interactive polling interface, and the TMDB Watch Providers API forstreaming availability filtering, the system delivers a holistic and fair group decision-making experience.

The prototype demonstrates that hybrid filtering outperforms single-method approaches, the choiceofaggregation strategy meaningfully affects group satisfaction, and SBERT mood filtering adds contextual relevance with 78% of sessions rating mood-filtered shortlists as more appropriate. The transparent pollingmechanism consistently improved perceived fairness and user engagement.

Future work will focus on scaling the user study, integrating neural collaborative filtering, adaptivemoodthreshold tuning, and extending the system to support television series and cross-session preference learning. Movie Bubble transforms the nightly group movie debate into a fun, shared, collaborative experience.

ACKNOWLEDGEMENT

The authors wish to thank their project guide, SK. Mujafar Ahmed, Assistant Professor, Department of AI &ML, Malla Reddy University, for his continued support and guidance throughout the development of thisproject. The authors also acknowledge the Department of AI & ML, School of Engineering, Malla ReddyUniversity, Hyderabad, for providing the necessary resources and infrastructure.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, “Context-Aware Recommender Systems,” in Recommender Systems Handbook, Springer, 2011, pp. 217–253.
[2] L. Baltrunas, T. Makcinskas, and F. Ricci, “Group Recommendations with Rank Aggregation and CollaborativeFiltering,” in Proc. ACM RecSys, 2010, pp. 119–126.
[3] R. Burke, “Hybrid Recommender Systems: Survey and Experiments,” User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331–370, 2002.
[4] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, “Neural Collaborative Filtering,” in Proc. WWW, 2017, pp. 173–182.
[5] Y. Koren, R. Bell, and C. Volinsky, “Matrix Factorization Techniques for Recommender Systems,” IEEEComputer, vol. 42, no. 8, pp. 30–37, 2009.
[6] J. Masthoff, “Group Recommender Systems: Combining Individual Models,” in Recommender Systems Handbook, Springer, 2011, pp. 677–702.
[7] M. O’Connor, D. Cosley, J. A. Konstan, and J. Riedl, “PolyLens: A Recommender System for Groups of Users,”inProc. ECSCW, 2001, pp. 199–218.
[8] M. J. Pazzani and D. Billsus, “Content-Based Recommendation Systems,” in The Adaptive Web, Springer, 2007, pp. 325–341.
[9] E. Quintarelli, E. Rabosio, and L. Tanca, “Recommending New Items to Ephemeral Groups,” in Proc. ACMRecSys, 2016, pp. 285–292.
[10] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” inProc. EMNLP, 2019, pp. 3982–3992.
[11] The Movie Database, “TMDb API Documentation,” 2024. [Online]. Available: https://developer.themoviedb.org